

La Metodología AOPOA

The AOPOA Methodology

Julián Rodríguez, Ing., Miguel Torres, MSc., Enrique González, PhD.

Pontificia Universidad Javeriana, Bogotá, Colombia

j-rodriguezm@javeriana.edu.co, metorres@javeriana.edu.co, egonzal@javeriana.edu.co

Recibido para revisión 26 de Marzo de 2007, aceptado 15 de Junio de 2007, versión final 31 de julio de 2007

Resumen—Este artículo presenta en detalle las fases de análisis y diseño de AOPOA, una metodología de programación orientada a agentes basada en una aproximación organizacional. Esta aproximación permite diseñar sistemas complejos descomponiéndolos en otros más simples. El enfoque organizacional soporta la descomposición iterativa y recursiva basada en el concepto de metas, y al mismo tiempo identifica interacciones entre las entidades que componen el sistema; en cada iteración un nivel organizacional es desarrollado. Durante la fase de análisis, las tareas y los roles son detectados. Durante la fase de diseño, las interacciones son caracterizadas y administradas a través de protocolos de cooperación. En la iteración final, la parametrización de roles es llevada a cabo, lo cual permite especificar eventos y acciones asociados a cada rol. Estos roles serán agrupados posteriormente en meta-agentes, que son los que finalmente serán desplegados en el ambiente dentro del que se desenvolverá el sistema multia-gente.

Palabras Clave—Programación Orientada a Agentes, Metodologías de Diseño de Agentes, Sistemas Multi-Agente.

Abstract—This paper presents in detail the analysis and design phases of AOPOA. A methodology for Agent Oriented Programming, based on an organizational approach. This approach allows the design of complex systems by performing iterative and recursive decomposition based in the goal concept, and at the same time identifies interactions between the entities inside the system; In each iteration a new organizational level is developed. During the analysis phase, tasks and roles are identified. During the design phase, all interactions are characterized and managed through cooperation protocols. In the final iteration, each role is parametrized, this allows the actual specification of events and actions associated with each role. These roles then are grouped in meta-agents, which will be instantiated inside the environment in which the system will work.

Keywords—Agent Oriented Programming, Agent Design Methodologies, Multi-Agent Systems.

I. INTRODUCCIÓN

EL concepto de agente racional apareció en el campo de la inteligencia artificial como un nuevo enfoque conceptual y práctico para diseñar y construir sistemas inteligentes. Los agentes racionales son vistos como unidades actuando en un

entorno particular para alcanzar un conjunto de objetivos bien definidos. En un enfoque más general, un agente es una entidad que encapsula datos y comportamientos y que es capaz de realizar una tarea de una manera autónoma y proactiva. Normalmente, los agentes no trabajan solos, sino en grupos para alcanzar sus objetivos; este grupo de agentes cooperativos forman un Sistema Multi-Agente (SMA) [3].

Construir un SMA implica una visión diferente para los procesos de análisis y diseño. En ésta, debe tomarse ventaja de las características intrínsecas de los agentes. Esta aproximación al diseño se adapta bien al proceso de subdividir un problema complejo en otros más simples, que son resueltos por entidades individuales, los agentes; además, ofrece la posibilidad de diseñar una solución modular que permite un manejo más estructurado y coherente de la complejidad global del sistema. Sin embargo, es claro que mientras se subdivide el sistema, múltiples problemas aparecen para lograr que los agentes trabajen juntos de manera cooperativa, para lograr los objetivos del sistema. De esta manera la metodología debe ofrecer métodos no solamente para asignar las responsabilidades a los agentes, sino también para identificar y administrar las interacciones entre ellos [4, 5, 7, 8, 9].

El desarrollo de metodologías eficientes para las fases de análisis y diseño de un sistema basado en agentes es un campo de investigación aún abierto. Este artículo presenta en detalle las fases de análisis y diseño de AOPOA, una metodología de programación orientada a agentes basada en una aproximación organizacional (Para una vista más detallada, remítase a González & Torres [2], y [10]). En esta aproximación, un SMA es percibido como una organización. Al igual que en otras aproximaciones, el manejo de la complejidad se consigue a través de la descomposición organizacional del sistema en partes más simples. El punto clave de la aproximación de AOPOA es que al mismo tiempo los agentes son diseñados de una manera estructurada y progresiva, y las relaciones entre ellos son establecidas y caracterizadas de manera automática.

II. AOPOA

La Aproximación Organizacional a la Programación Orientada a Objetos (AOPOA) es una metodología para el análisis y diseño de Sistemas Multi-Agente (SMA). En ella, cada conjunto mayor de actividades está formado por un conjunto de procesos iterativos. Cada uno de esos procesos genera un conjunto de artefactos que incluyen tablas y diagramas que definen un SMA genérico.

Algunos de los conceptos que rodean la metodología fueron mencionados en la sección anterior; sin embargo, con propósitos de referencia, en el contexto de un SMA el término objetivo es equivalente en significado a la palabra meta.

El foco en esta sección es la presentación de los conceptos y procesos generales que son aplicados durante el uso de la metodología.

A. Enfoque Organizacional

Al analizar una organización se observa que ésta puede ser descompuesta en unidades bien definidas, cada una de las cuales tiene una responsabilidad clara en relación a las metas de la organización. Con el fin de lograr que los diferentes componentes de la organización puedan cooperar en el cumplimiento de objetivos compartidos y logren una gestión correcta de los recursos es indispensable que estos componentes interactúen. La descomposición y la gestión de las interacciones, son los elementos distintivos que soportan conceptualmente a AOPOA.

Típicamente, al analizar una organización se encuentra que está compuesta de un conjunto de entidades autónomas, que actúan proactivamente para alcanzar los objetivos de la organización. Normalmente, las entidades están agrupadas en equipos pequeños de trabajo, los cuales a su vez pueden ser agrupados de manera recursiva, hasta llegar a percibir a toda la organización como si fuera un único equipo compuesto de los equipos de más alto nivel. Realizando el mismo proceso, pero en sentido contrario, es posible sintetizar una organización haciendo una descomposición recursiva. En efecto, la organización se divide en entidades abstractas de alto nivel, las cuales se subdividen en forma recursiva hasta llegar a las entidades autónomas que la componen.

La Figura 1 ilustra estos conceptos; como se puede observar se genera una jerarquía organizacional, la cual puede ser modelada como un árbol. Nótese que no se requiere que el árbol sea balanceado.

En AOPOA, un rol representa una entidad abstracta que debe alcanzar un conjunto de objetivos mediante el aprovechamiento de sus habilidades y utilizando los recursos disponibles en su entorno. En consecuencia, un SMA puede ser diseñado como una jerarquía de roles, los cuales pueden ser obtenidos mediante un proceso de descomposición basado en los objetivos. El árbol de roles que representa la jerarquía organizacional puede desarrollarse de diferentes formas, la selección del orden en que se descomponen los roles existentes puede basarse en un recorrido en amplitud ó en profundidad. Una alternativa más interesante, puede ser descomponer primero los roles más complejos.

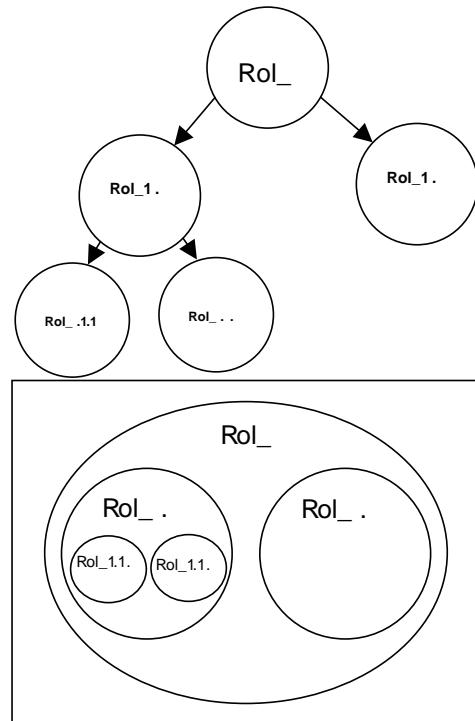


Figura 1. Descomposición recursiva de una organización.

En el proceso de análisis o el de síntesis, para lograr un adecuado funcionamiento es imperativo que las entidades interactúen entre si. En muy pocos contextos podría ser de verdadera utilidad tener una organización compuesta por entidades independientes y separadas. Una característica muy importante de las interacciones es que éstas surgen en situaciones típicas, cuándo varios roles aportan a la consecución de un objetivo. Primero, se requiere establecer qué tarea va a cumplir cada uno de los roles involucrados; en este contexto, aparecen interacciones para realizar el manejo de la asignación de tareas. Segundo, se precisa coordinar el accionar de los participantes, pues el orden en que se realizan las acciones puede ser importante para obtener el resultado deseado; en este caso, emergen interacciones para lograr la planificación y la ejecución sincronizada de las acciones. Tercero, se deben detectar y resolver los conflictos que puedan aparecer en el acceso a los recursos compartidos; en estas situaciones, se generan interacciones para realizar un acceso controlado a los recursos. Nótese que al analizar estos tres casos, los vínculos de cooperación aparecen siempre que hay un objetivo compartido por varios, ó cuando hay recursos en común. En AOPOA, se aprovecha esta característica implícita de las interacciones para detectar en forma automática los potenciales vínculos de cooperación entre roles que pueden ser requeridos para alcanzar los objetivos del sistema.

Los dos conceptos anteriormente analizados, la descomposición organizacional y la detección de vínculos de cooperación, están fuertemente ligados. En efecto, AOPOA propone realizar una descomposición gradual en roles; tomar un rol muy complejo y dividirlo en varios más simples, pero al

mismo tiempo identificar si entre dichos roles resultantes surgen vínculos de cooperación.

Para realizar la identificación de los roles derivados, AOPOA propone una heurística que propende por la especialización de los roles y por la disminución del número de interacciones. La especialización es una característica deseable, que permite una mayor cohesión de los componentes resultantes; para lograrla, la heurística favorece la descomposición en la que los roles no comparten las mismas habilidades. La reducción del número de vínculos reduce el nivel de comunicaciones entre roles; la heurística penaliza la aparición de recursos compartidos.

III. FASE DE ANÁLISIS

En esta fase, una vez se ha determinado que el problema posee las características en las cuales una solución basada en

agentes es adecuada, se realizan dos procesos: la caracterización general del sistema y la descomposición organizacional recursiva.

A. Caracterización del Sistema

La dinámica del proceso de caracterización general del sistema es ilustrada en la Figura 2. Al final de la realización de las actividades asociadas se obtiene el *Diagrama de Vínculos Externos* del sistema, el cual modela las relaciones de éste con las entidades externas que se encuentran en el ambiente de trabajo. Adicionalmente, se genera la primera versión de los artefactos *Tabla de Roles* y *Tabla de Tareas*, los cuales se constituirán en el artefacto central del proceso recursivo de descomposición organizacional.

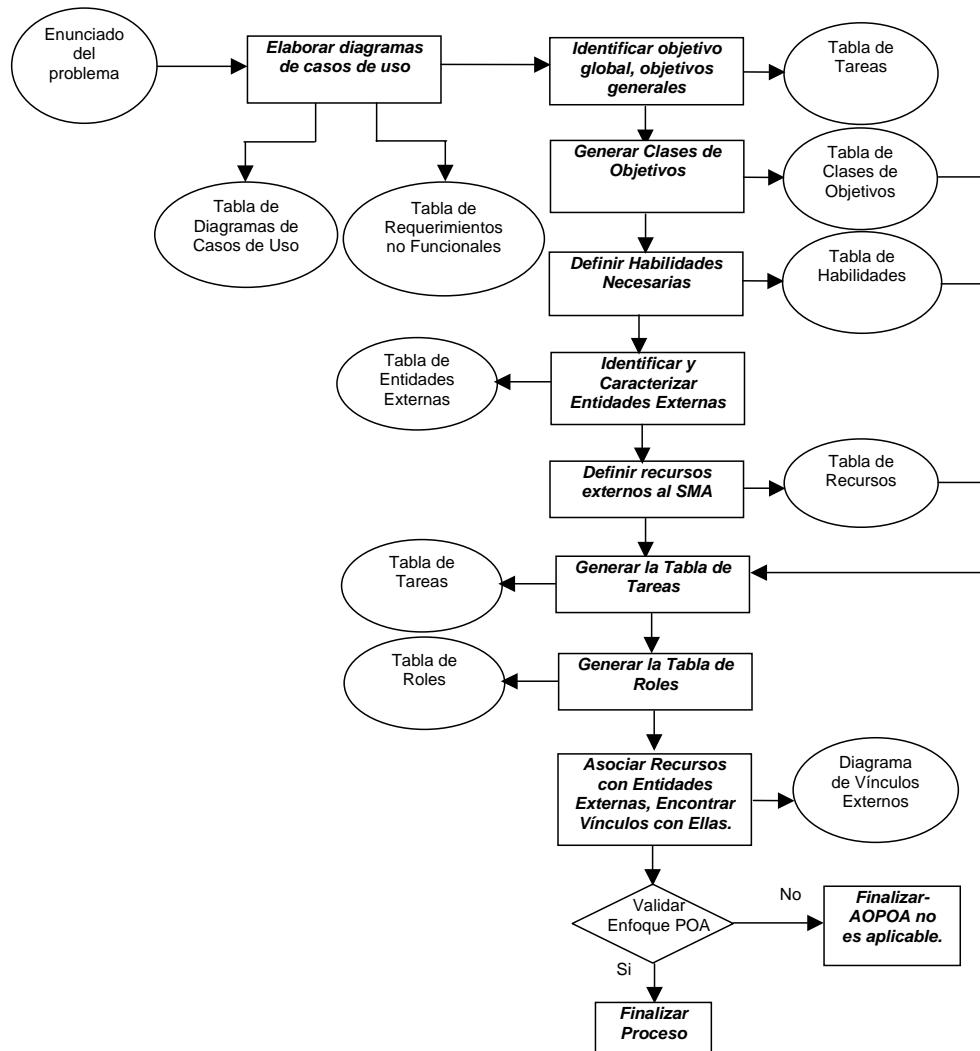


Figura 2. Descomposición recursiva de una organización

La primera actividad es la identificación de los objetivos generales a partir del análisis de requerimientos del sistema. Estos objetivos constituyen una aproximación general de la

solución, representan las responsabilidades del sistema como si éste estuviese constituido por un rol, el cual será refinado durante el proceso de descomposición organizacional.

Cada objetivo tiene una clase asociada, y la lista de estas clases se consigna en la *Tabla de Clases de Objetivos*. Para cada uno de los objetivos se deben definir las habilidades y los recursos requeridos para su realización; la información definida en cada caso, se consigna respectivamente en la *Tabla de Habilidades* y la *Tabla de Recursos*. Las tres tablas mencionadas (Clases de Objetivos, Habilidades y Recursos) conforman un único artefacto llamado *Artefacto Descriptor de Tareas*, cuya información es utilizada para generar la *Tabla de Tareas*, en la que cada objetivo da lugar a una tarea. En este punto, el SMA se modela por un único rol, el cual tiene a su cargo todos los objetivos globales del sistema; este rol será la

primera entrada en la *Tabla de Roles*.

A continuación, se identifican y caracterizan todas las entidades externas con las cuales el SMA deberá interactuar. Existen dos tipos de entidades externas: las entidades externas activas son denominadas actores, y las pasivas son los objetos que conforman parte del ambiente. En ambos casos, estas entidades serán asociadas por el SMA con los recursos requeridos. En el primer caso, los recursos pueden ser obtenidos a través de los vínculos con los actores externos. En el segundo caso, los sensores y actuadores de los agentes son utilizados para percibir y manipular los objetos pasivos.

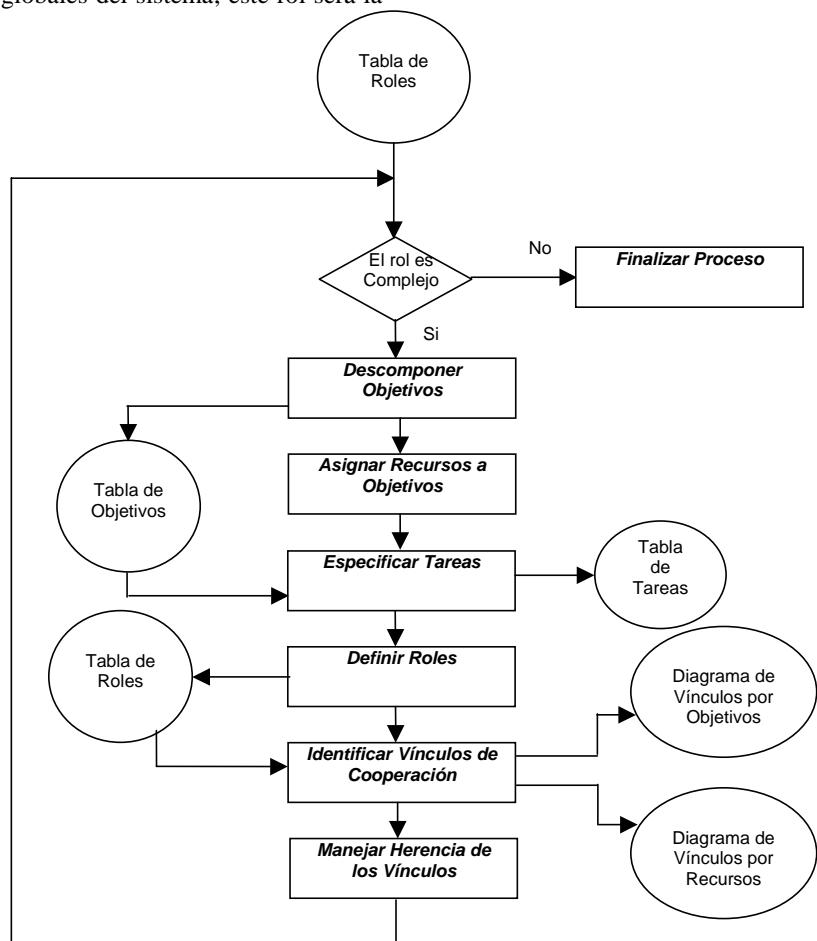


Figura 3. Proceso de descomposición organizacional

B. Descomposición Organizacional

Una vez han sido identificados y caracterizados las entidades externas, los recursos y los objetivos generales del sistema, se debe proceder a realizar el proceso recursivo de descomposición organizacional en roles. En la primera iteración, el sistema aparece representado por un único rol; en las iteraciones siguientes, ya se habrá iniciado el desarrollo del árbol jerárquico de roles. En cualquier caso, la primera actividad es evaluar la complejidad de los roles que constituyen la hojas del árbol. Si se determina que la complejidad de uno de los roles es muy alta, éste debe ser subdividido.

En cada iteración se efectúa la descomposición recursiva de un rol complejo en dos o más roles simples. El proceso inicia con la descomposición de los objetivos del rol complejo en objetivos más simples. Estos objetivos simples son analizados con el fin de identificar los recursos y habilidades que se requieren para su obtención. Cada objetivo se asocia a una tarea; es ésta la que incorpora en una unidad lógica un objetivo con las clases de objetivo, las habilidades y los recursos. La Tabla de Tareas es modificada para reflejar estos cambios.

La siguiente actividad tiene como fin obtener una agrupación de tareas; es decir, un conjunto de grupos de

tareas. Cada uno de los grupos obtenidos constituye la base de uno de los nuevos roles simples derivados del rol complejo. La mejor agrupación es producida mediante la aplicación de una heurística. La idea es seleccionar la agrupación que tienda a favorecer la generación de roles que compartan el mínimo de recursos, que tengan la menor redundancia de habilidades, y que incorporen en lo posible la mayoría de los objetivos de una misma clase.

La definición de roles efectuada a partir de la mejor agrupación de tareas es registrada en la *Tabla de Roles*. Luego, una vez se han definido los nuevos roles, se procede a la actividad de identificación de vínculos de cooperación entre éstos. Estos vínculos pueden surgir por dos motivos:

- los nuevos roles utilizan recursos compartidos.
- los nuevos roles contribuyen a la obtención de un objetivo común.

Antes de finalizar la iteración, también debe analizarse la herencia de los vínculos que ya poseía el rol complejo. Estos vínculos de cooperación deben ser asumidos por alguno de los nuevos roles identificados.

El proceso de identificación de roles es llevado a cabo de manera iterativa cuantas veces se juzgue necesario, subdividiendo en cada paso uno de los roles encontrados. Una vez concluido este proceso, es posible pasar a la etapa de diseño.

C. Etapa de Diseño

En la etapa de análisis se realizó un modelo del comportamiento del sistema concentrándose en qué roles deben existir y qué es lo que éstos deben hacer. En la etapa de diseño, se refina el modelo realizado durante la etapa de análisis, pasando de una vista global a una vista local, en la que basándose en los conceptos de SMA, se analizan las necesidades de cooperación e interacción entre los roles encontrados. En la etapa de diseño se llevan a cabo dos procesos: el diseño de vínculos de cooperación y el diseño de agentes.

1) Diseño de Vínculos de Cooperación

La dinámica del proceso de diseño de vínculos de cooperación se muestra en la. Al finalizar el proceso se obtendrá el Diagrama de Vínculos de Comunicación, que muestra los mensajes que surgen entre los roles encontrados en el SMA, generados por los vínculos de cooperación entre ellos.

A continuación se caracterizan como interacciones los vínculos de cooperación detectados durante el análisis, identificando para cada uno el tipo de interacción; por ejemplo: independencia, cooperación o antagonismo. El tipo de interacción se asigna de acuerdo con la información obtenida de la *Tabla de Recursos*, el *Diagrama de Vínculos por Recursos*, y el *Diagrama de Vínculos por Objetivos* para cada caso. El producto de esta caracterización es registrado en la *Tabla de Tipos de Interacción*.

Una vez se ha completado la *Tabla de Tipos de Interacción*, se procede a identificar una técnica de cooperación adecuada;

para cada caso se determina si el manejo de la interacción requiere el uso de mecanismos de colaboración, coordinación ó resolución de conflictos. Cabe anotar que en algunos casos, este manejo puede requerir el uso de mecanismos que realicen más de una de estas tres funcionalidades. La selección de la técnica o mecanismo de cooperación particular a utilizar queda a criterio del diseñador, por cuanto existen numerosas alternativas para ello.

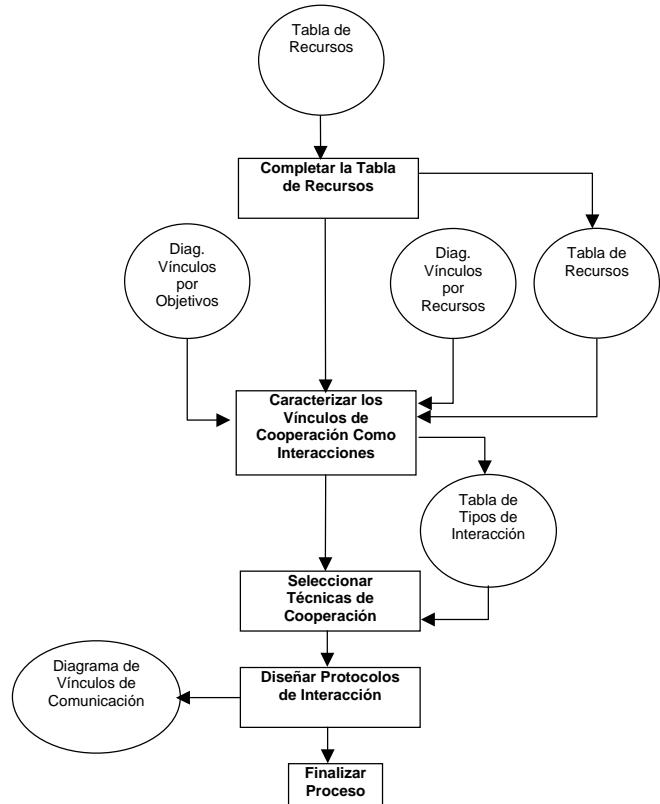


Figura 4. Diseño de vínculos de cooperación

Por último, para cada vínculo de cooperación, a partir de las técnicas seleccionadas en la actividad anterior, se diseñan protocolos de interacción que permitan manejar de manera apropiada cada caso de interacción entre roles. El diseño de un protocolo implica determinar el flujo específico de eventos que se intercambian entre los roles involucrados, así como también la semántica y los datos asociados a cada uno de estos eventos. Es posible recurrir a protocolos predefinidos, como los propuestos por FIPA [1], o crear nuevos protocolos que se ajusten a las necesidades del diseñador, y del problema. Para este caso, pueden incluso definirse "roles sociales" o auxiliares que se encarguen de ser mediadores o facilitadores en el manejo de ciertas situaciones de interacción. La especificación detallada de los protocolos y de los eventos se consigna en el *Diagrama de Vínculos de Comunicación*.

2) Diseño de Agentes

Una vez se han detallado todos los vínculos de cooperación, sigue el diseño interno de los roles identificados. La dinámica de este proceso se muestra en la Figura 5.

Inicialmente se deben especificar de un modo más detallado los roles encontrados hasta el momento. Con este fin, deben definirse de manera específica los eventos a los que cada rol responderá. Cada evento, proveniente del exterior del rol provocará una respuesta por parte del mismo, en la que estarán involucrados el estado del agente, los recursos y las habilidades. La información básica que describe a cada evento es registrada en la *Tabla de Eventos*.

La siguiente actividad contempla la definición de los componentes que describen el estado de un rol. Éstos son la memoria del rol, y le brindan la información que necesita tener para poder reaccionar adecuadamente a los eventos que le sean comunicados. La descripción detallada del estado es almacenada en la *Tabla de Estados*.

Como parte del proceso de respuesta a eventos, el rol debe percibir y manipular los recursos disponibles en el ambiente; estas tareas son llevadas a cabo utilizando adaptadores. Cada adaptador debe tener asociada al menos una primitiva que proporcione los medios para llevar a cabo las funcionalidades requeridas para controlar uno o varios recursos. La definición de los adaptadores y sus primitivas asociadas es consignada en la *Tabla de Adaptadores*.

A continuación, se debe proceder a especificar de una manera más detallada las respuestas del rol a los eventos previamente definidos. La información referente a esta cuestión se registra en la *Tabla de Eventos*. Se incluye aquí la habilidad mental involucrada en cada respuesta. Una habilidad mental es el mecanismo por medio del cual el rol toma la decisión de comportarse de una u otra manera como respuesta a un evento.

Como última actividad del diseño interno de los roles debe revisarse el cumplimiento de los requerimientos no funcionales definidos para el SMA durante la etapa de análisis. En caso de ser necesario, deben modificarse los componentes de estado, las habilidades mentales, y los demás componentes del rol que se juzgue oportuno.

Con los roles completamente definidos, es el momento de agruparlos en meta-agente. Un meta-agente es un espacio de ejecución para los roles, que puede agrupar uno o más de ellos. El meta-agente asume el cumplimiento de los objetivos de todos los roles que lo integran, respondiendo en la forma especificada para cada rol, a los eventos asociados. Los criterios de agrupación de roles en meta-agente son variados, se puede tener en cuenta la similitud de objetivos, la distribución de habilidades, o el acceso a los recursos. Se debe analizar la granularidad de la agrupación resultante, en términos de la complejidad de los roles agrupados en un meta-agente determinado. Los meta-agentes resultantes serán instanciados en agentes que contendrán de manera lógica y física todas las características de los roles definidos durante las etapas de análisis y diseño.

Una vez han sido obtenidos los meta-agente que agrupan los roles definidos durante el análisis y diseño del sistema, es necesario llevar a cabo la etapa de Análisis de Despliegue, en la que se caracteriza el ambiente en el que los agentes reales estarán ubicados y operando para cumplir con los objetivos

del sistema. Esta caracterización incluye la definición de sitios de despliegue, y para cada sitio, la disponibilidad de recursos, almacenamiento, y capacidad de cómputo, entre otros. Teniendo esta información definida, se procede a verificar las modificaciones que hay que hacer a los meta-agentes para adaptarse al ambiente de despliegue, como por ejemplo replicación o distribución de instancias. Todos estos aspectos son relevantes al momento de tomar decisiones en términos de implementación del sistema, y claramente durante el despliegue físico del mismo.

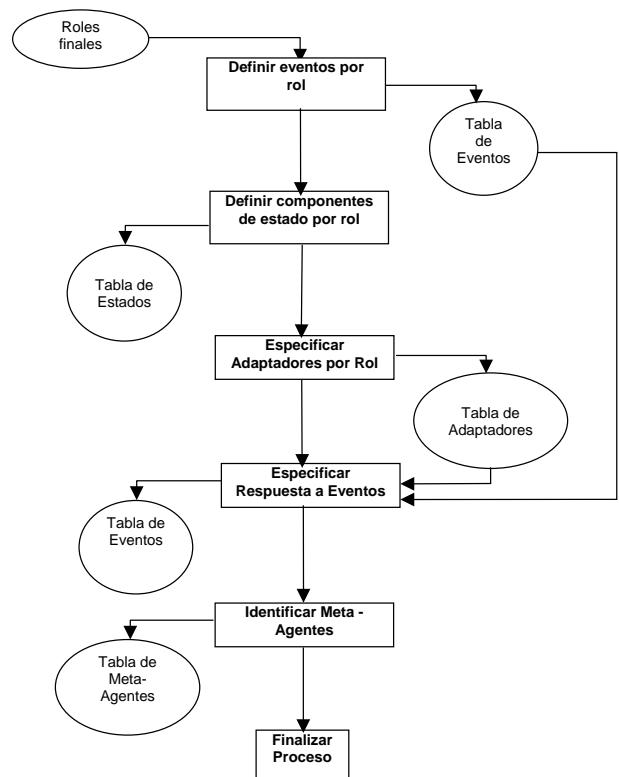


Figura 5. Diseño de agentes.

IV. CASO DE ESTUDIO

Como caso de estudio de aplicación de la metodología AOPOA a un problema del mundo corporativo, ésta ha sido aplicada al problema de administración y optimización de servidores en la Empresa de Telecomunicaciones de Bogotá ETB. Los servidores de negocios de la compañía han venido siendo administrados utilizando el software HP OpenView. Sin embargo, varios factores incluyendo costos de licenciamiento, consumo de recursos, despliegue estático y proactividad limitada, hicieron deseable la implementación de un sistema paralelo de gestión, diseñado siguiendo una aproximación multi-agente.

El proceso inició con el levantamiento de información correspondiente al problema. Dicho levantamiento fue llevado a cabo siguiendo los procedimientos estándares de levantamiento de información para proyectos de software no enfocados utilizando el paradigma multi-agente. A partir del resultado de este proceso se planteó el objetivo principal de la

solución de software a implementar, se identificaron los actores principales, y se definieron los casos de uso de contexto que son necesarios para comenzar a aplicar la metodología AOPOA en todas sus fases.

Una vez lista la información inicial necesaria para aplicar la metodología, se procedió a ello, utilizando una herramienta CASE específicamente desarrollada para aplicar AOPOA. Utilizando dicha herramienta llamada AOPOA CASE Tool, y tomando como base los requerimientos funcionales y no funcionales levantados durante el proceso inicial, y los diagramas de casos de uso luego generados, se llevó a cabo la secuencia de etapas definidas por la metodología en sus fases de análisis y diseño. El resultado final de este proceso fue un sistema multi-agente orientado a suplir las necesidades de gestión de servidores detectadas durante el levantamiento de información.

Para proceder a la implantación de la solución diseñada utilizando la herramienta CASE, se utilizó una de las características más útiles de dicha herramienta: la generación automática de código. Utilizando esta característica, es posible partir del diseño del sistema multi-agente realizado con AOPOA CASE Tool, y generar un conjunto de archivos de código Java que se integra automáticamente con el framework BESA (Behavior-oriented, Event-driven and Social-based Agent) [6] para ofrecer una solución integral al problema planteado inicialmente. Dichos archivos contienen la base del sistema, y luego de ser completados en términos de funcionalidades específicas, utilizan las facilidades ofrecidas por el BESA para dar como resultado la aplicación multi-agente final a ser implantada en la compañía.

Con el código de la solución multi-agente completo, la fase final del proceso fue la implantación en la ETB. Para este fin, se contó con un conjunto de servidores de pruebas designados por la compañía. Una vez instalada la solución, se procedió a aplicar un protocolo de pruebas definido en conjunto con los responsables técnicos de la operación de los servidores en la ETB, y orientado a la estimación de la utilidad general e idoneidad en tiempos de respuesta de la solución multi-agente propuesta. Presentar en detalle el protocolo de pruebas y los resultados detallados del mismo se encuentra fuera del alcance de este artículo. Sin embargo, el desempeño de la solución fue tan satisfactorio que produjo la intención de extender su instalación a otros servidores de la compañía en un futuro.

V. CONCLUSIONES Y TRABAJO FUTURO

AOPOA está basada en una aproximación organizacional, que permite realizar una descomposición orientada a las metas y la detección de enlaces cooperativos entre los roles organizacionales. La heurística propuesta reduce las interacciones entre roles, evita los conflictos por recursos, reduce la redundancia en las habilidades asociadas a cada rol, y toma en cuenta el conocimiento previo del problema. Otra característica importante de AOPOA es la eficiencia en la identificación de roles, ya que la descomposición es realizada de manera gradual. AOPOA genera una solución sub-óptima

pero que basada en el uso de la heurística propuesta asegura la calidad y facilidad de administración del sistema. La identificación temprana y automática de los enlaces cooperativos debidos a recursos compartidos, habilidades complementarias y metas comunes permite manejar la complejidad, y genera un conjunto consistente de relaciones entre los roles al final del proceso. Ya que las actividades propuestas son completamente sistemáticas, una herramienta CASE fue desarrollada para apoyar el uso de la metodología y diversos proyectos en clases han sido desarrollados utilizando AOPOA, con el fin de probarla y refinirla.

Trabajos futuros en esta área estarán enfocados en el refinamiento de la metodología AOPOA y su herramienta CASE asociada. En particular, las actividades asociadas al proceso de pruebas del desarrollo de software serán mejoradas. Con el fin de validar AOPOA en la construcción de SMAs compuestos de agentes físicos, se estudiará el problema de la implementación de soluciones basadas en agentes cooperativos aplicadas a robótica móvil utilizando la metodología.

VI. AGRADECIMIENTOS

Los miembros del equipo de investigación, quieren expresar su agradecimiento a COLCIENCIAS y la ETB por el apoyo prestado durante el proceso de refinamiento y utilización de la metodología AOPOA. Los resultados reportados en este artículo forman parte del proyecto de investigación “AGS-AOPOA: Agentes para la Gestión de Servidores”. Los autores agradecen los aportes realizados por los miembros del equipo de trabajo; en especial, a Diana Ahogado y Ana María Reinemer por su aporte en la formulación inicial de la metodología AOPOA.

REFERENCIAS

- [1] The Foundation for Intelligent Agents <http://fipa.org> – Fecha de acceso: Enero 12 de 2006.
- [2] Gonzalez E., Torres M. “AOPOA Organizational Approach for Agent Oriented Programming”, in Proceedings of the 8th International Conference on Enterprise Information Systems, May 2006 Paphos – Cyprus
- [3] Ferber, Jacques. “Multi-agent Systems: An Introduction to Distributed Artificial Intelligence”, Addison – Wesley Longman. 1ra. Ed., 1999.
- [4] Wooldridge M., Jennings N. “Agent Technology – Foundations, Applications, and Markets”. Springer - UNICOM. 1998.
- [5] Wooldridge, Michael. “Reasoning about rational agents”. The MIT Press (Cambridge, Massachusetts/London, England) as part of the Intelligent robotics and autonomous agents series. 2000.
- [6] González, E., Bustacara, C., Ávila, J. (2003). “BESA: Behavior-oriented, Event-driven and Social-based Agent Framework”. PDPTA '03.
- [7] Wooldridge M., Jennings M., Kinny D. The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems, 3, 285@312, 2000 Kluwer Academic Publishers. Manufactured in The Netherlands.
- [8] <http://troposproject.org/> , “Tropos – Requirement-Driven Development for Agent Software”. Universidad de Trento – Última acceso: 12 de Enero de 2006.
- [9] Torres da Silva, V., Choren, R., Lucena , C. “A UML Based Approach for Modeling and Implementing Multi-Agent Systems”, ACM, 2004.
- [10] González E., Bustacara C. “Desarrollo de Aplicaciones Basadas en Sistemas Multi-Agentes”. Libro Producto de Investigación – Grupo

SIDRe, Departamento de Ingeniería de Sistemas, Facultad de Ingeniería,
Pontificia Universidad Javeriana.

Julián E. Rodríguez Asistente de Investigación del grupo SIDRe, Pontificia Universidad Javeriana, Departamento de Ingeniería de Sistemas. Ingeniero de Sistemas graduado de la Universidad Nacional de Colombia, es un asistente de investigación en la Pontificia Universidad Javeriana, principalmente en el campo de los sistemas Multi - Agentes. Con más de 10 años de experiencia profesional en los campos de diseño y desarrollo de software, trabajos previos incluyen investigación en el campo de los fractales y el modelado de elementos naturaleza utilizando geometría fractal.

Miguel E. Torres Moreno Ingeniero de Sistemas egresado de la Universidad Nacional de Colombia Sede Bogotá, graduado en el año 1999. Su tesis se tituló: "Diseño de Circuitos Combinacionales por Medio de Programación Genética" la cual obtuvo una mención meritoria. Se desempeñó como Ingeniero de Desarrollo en la empresa Neotech Engineering Ltda. (1999-2001) en diferentes proyectos como fueron: Procesador Integral de Transacciones Banco Agrario de Colombia, Traductor POS Subsidiado CAFESALUD S.A., Software de soporte para el Programa Familias en Acción de la Presidencia de la República y Facturación POS Subsidiado CAFESALUD S.A. Obtuvo su título de Master in Computer Sciences de Mississippi State University (2002-2003) donde además se desempeñó como Asistente de Investigación en The Center for Computer Security Research donde se realizó tareas de análisis y diseño de herramientas para la detección de intrusos en estaciones de trabajo, detección de comportamiento anómalo de programas UNIX y generación de ataques simulados a un High Performance Computer Cluster con sistema operativo Linux. Se desempeña como Profesor Asistente en el departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana.

Enrique González Guerrero Ingeniero Eléctrico, Universidad de los Andes. Magíster en Ingeniería Eléctrica, Universidad de los Andes. D.E.A. en Robótica, Universidad Paris VI – Francia. Doctorado en Informática, Universidad Evry – Francia. Investigador y docente de la Pontificia Universidad Javeriana desde 1999, director del grupo de investigación SIDRe del Departamento de Ingeniería de Sistemas. Su trabajo se ha enfocado en la integración de los Sistemas Distribuidos y la Inteligencia Artificial, confluyendo hacia el desarrollo del proyecto marco Arquitectura para Sistemas Multi - Agentes, el cual ha generado conocimiento y herramientas para el desarrollo de aplicaciones basadas en agentes, la programación orientada a agentes y la robótica cooperativa. Los modelos conceptuales desarrollados y las herramientas que se han generado constituyen un marco para el análisis y síntesis de sistemas complejos.