

Sistema multi-agente para el apoyo a la gestión de inventarios en itil mediante el monitoreo distribuido de software y hardware en una red corporativa

Multi-agent system to inventories management support in itil through the distributed hardware and software monitoring in a corporate network

Sebastián Ortiz.¹, Cesar López Gallego, I.S.² y Ana Isabel Oviedo Carrascal, M.Sc.³

1. Estudiante Ingeniería Informática, 2. Docente de Ingeniería Informática, 3. Docente de Ingeniería Informática
Universidad Pontificia Bolivariana, Medellín-Colombia
{sebasor, calopezg, anaisaovaldo}@gmail.com

Recibido para revisión 28 de Septiembre de 2008, aceptado 25 de Octubre de 2009, versión final 28 de Noviembre de 2009

Resumen—Gestionar la información del inventario de software y hardware puede ser, para una organización, una tarea dispendiosa y a veces no alcanzable por la complejidad y el número de usuarios que en ella coexisten. Como un acercamiento a la solución de este problema, se plantea el diseño de un sistema multi-agente distribuido que implemente un monitoreo constante en tiempo real y usando la red corporativa, de los cambios en el software y hardware de las máquinas de los usuarios.

Palabras Clave—Gestión de Inventarios, ITIL V3, Sistemas Multiagentes, Metodología MAS-CommonKADS.

Abstract—Information Management of software and hardware inventory can be, to an organization, a wasteful task and not attainable, because the complexity and number of users. An approach to solve this problem, we are designing a distributed multi-agent system that implements in real time a constant monitoring of changes in software and hardware of users' machines in a corporate network.

Keywords—Inventories Management, ITIL, Multi-agent System, MAS-CommonKADS Metodology.

I. INTRODUCCIÓN

Es común encontrar en las organizaciones actuales, una alta demanda de su personal por dispositivos de cómputo en sus sitios de trabajo. Los equipos de cómputo en una organización crecen casi en la misma proporción con el crecimiento del personal y es necesario que los responsables de

los procesos de TI (Tecnología Informática) creen políticas para el uso de estos y tengan mecanismos como modelos y herramientas, que les ayuden a velar por su adecuado cumplimiento [20].

La gestión de inventarios es un proceso de los modelos de gestión de TI, la cual tiene como objetivo mantener actualizada la información de la configuración de hardware y software de los recursos de cómputo que se encuentren en los escritorios de los usuarios finales [12].

El resultado de la gestión de inventarios contribuye a respaldar las políticas de seguridad del proceso de TI en una organización que tienen como fin evitar la instalación de software no licenciado en las estaciones de trabajo de sus usuarios. Actualmente, es fácil para un usuario instalar copias de software no licenciado en el equipo de cómputo de su escritorio. Algunas formas a las que tradicionalmente acuden son la descarga de instaladores desde múltiples servidores de internet; el acceso fácil a medios con múltiples instaladores que se consiguen con vendedores de PC's y distribuidores "piratas"; cracks con seriales para legalizar copias trial o copias no legales y programas tipo torrent para descargar de forma rápida archivos de gran tamaño.

Este artículo propone el diseño y desarrollo de un sistema multiagente, que de manera automatizada, ayude a los responsables del proceso de Gestión de Inventarios en cualquier organización, a garantizar el cumplimiento sobre los acuerdos de software legal incluido en las políticas de uso de computadores de escritorio. En la sección 2 se describe el área de Gestión de Inventarios. En la sección 3 se presenta un marco conceptual de Sistemas Multiagentes. En la sección 4 se presenta un estado

del arte en metodologías de desarrollo de Sistemas Multiagente. En la sección 5 se describe la solución propuesta. En la sección 6 se describe la evaluación y finalmente en la sección 7 las conclusiones y trabajo futuro.

II. GESTION DE INVENTARIOS EN ITIL

El proceso de gestión de inventarios puede ser definido como la captura de información, es decir, de la disponibilidad, responsabilidad, localización y estado actual de los activos que componen la infraestructura de TI en una organización. La meta es tener una completa, adecuada y actualizada vista de todos los componentes de una red, incluyendo PCs, servidores, impresoras, concentradores, routers, switches y software, que componen la infraestructura de TI. Como mínimo este proceso debe proporcionar información de la clase de dispositivos que están instalados. En un momento de tiempo dado, el proceso debe entregar el estado actual de todos los componentes de la infraestructura [10].

La automatización de la gestión de inventarios presenta diversos beneficios, como ahorrar tiempo y esfuerzo para llevar a cabo los inventarios físicos; reducir la interrupción en el trabajo de los usuarios en la actividad de toma de información para inventarios; tener una visión sistemática del contenido de la configuración de un activo al momento de analizar la ocurrencia de un incidente; optimizar el uso de los recursos de hardware y software en la organización adecuándolos a las necesidades reales; facilitar las labores de auditoría proporcionando una visión general de los componentes de la red; mejorar los procesos de compra de activos mediante la confrontación de la capacidad actual con la capacidad requerida de una implementación nueva; reducir el riesgo de incurrir en multas y/o procesos legales por el uso de software ilegal; y controlar cambios en piezas hardware [20].

También se pueden definir algunas buenas prácticas en la gestión de inventarios, como procurar que el proceso esté bien definido y que el inventario sea mantenido de forma adecuada; establecer políticas de frecuencia en las actividades de búsqueda para obtener información; implementar comparativos del estado de la configuración entre búsquedas; asegurar que los usuarios de la organización tengan reportes periódicos utilizar un repositorio central como la CMDB (Configuration Management Database) donde permanecen almacenados los ítems de inventario; complementar con modelos (por ejemplo de minería de datos) que permitan explorar los ítems del inventario tan profundo o tan ancho como sea necesario; y definir un responsable del proceso [11].

En la actualidad, existe un framework público que describe una "buena práctica" para la gestión de servicios de TI, llamado ITIL, el cual se enfoca en la medición y el mejoramiento continuo de la calidad de entrega de los servicios de TI desde la

perspectiva de una organización y sus clientes. Su aceptación a nivel mundial se debe a los beneficios que han obtenido las organizaciones que han seguido el modelo para gestionar sus servicios de TI, entre estos beneficios se encuentran un incremento de la satisfacción de usuarios y clientes con los servicios; el mejoramiento en la disponibilidad del servicio que incide en mejores beneficios e ingresos; beneficios financieros derivados de la identificación y reducción de trabajos repetidos, tiempos perdidos y el mejoramiento de la gestión y uso de los recursos; y el mejoramiento de los procesos de análisis de riesgos [18].

ITIL se publicó en el Reino Unido entre 1989 y 1995 por la HMSO (Her Majesty's Stationery Office) en nombre de la CCTA (Central Communications and Telecommunications Agency) - actualmente inmerso en la OGC (Office of Government Commerce). Actualmente está vigente la versión 3 publicada en Mayo de 2007 y su núcleo esta compuesto por 5 libros que cubren el ciclo de vida de los procesos de tecnología; los libros son: Estrategia de Servicios, Diseño de Servicios, Transición de Servicios, Operación de Servicios y Mejoramiento Continuo de los Servicios. La gestión de inventarios en la versión actual es un elemento del proceso de gestión de configuración y activos enmarcado en la fase de transición de servicios [11].

En ITIL, el proceso de gestión de inventarios almacena la información de los componentes en un repositorio central, que en el caso de ITIL se denomina CMDB (Configuration Management Database), a través del cual se soportan otros procesos como service delivery, service support, IT asset Management; los cuales son complementados con información de incidentes, errores conocidos y cambios de versiones. La implementación de un proceso de gestión automatizado es crítico para los procesos de gestión de configuración, de gestión de activos de TI y las disciplinas de gestión de servicios.

A los procesos de gestión de configuración y gestión de activos, la gestión de inventarios le agrega valor mediante la adición de relaciones dinámicas entre los ítems del inventario y eventualmente con otros registros almacenados en la CMDB.

III. SISTEMAS MULTIAGENTES

Para realizar una conceptualización del área de sistemas multiagente, en adelante SMA, es necesario introducir la definición de un agente; aunque no se tiene una definición exacta, un agente puede ser definido como un proceso computacional que implementa funciones autónomas y comunicativas, se comunica usando un lenguaje de comunicación de agentes, combina una o más capacidades de servicio, actúa en nombre de un humano o una entidad y soporta al menos una noción de identidad conocida como AID (Agent Identifier) que marca un agente para que pueda ser distinguido sin ambigüedad en el universo de agentes [8].

Los SMA son un conjunto de agentes autónomos con tres características: organización, coordinación y comunicación. La organización establece una jerarquía social de los agentes. La coordinación puede ser por cooperación y/o competición. La comunicación se realiza por medio de protocolos que permiten estructurar la interacción entre agentes.

Aunque son similares, los agentes tienen características que no poseen los objetos como autonomía, cooperación, percepción y pro-actividad. La principal diferencia es que los objetos son pasivos, reaccionan a estímulos externos, pero no tienen metas que direccíonen su comportamiento, como los agentes. Otra diferencia es que los agentes usan un lenguaje común entre todos los agentes, mientras que los mensajes entre los objetos dependen de las clases [22].

La construcción de SMA integra tecnologías de distintas áreas de conocimiento. Por un lado la Inteligencia Artificial [22] se ha ocupado de dar comportamientos inteligentes a los agentes basándose en actitudes como: racionalidad, coherencia y capacidad de adaptación. Por otro lado la Ingeniería de Software [7] ha apoyado el desarrollo de metodologías orientadas a agentes por su relación tan cercana a la tecnología del objeto y a las metodologías orientadas al objeto. El término adoptado para estas metodologías es AOSE - Agent-Oriented Software Engineering. Las dos áreas mencionadas han dado gran impulso al desarrollo de SMA, promoviendo un proceso de formación de la computación orientada a agentes.

La computación orientada a agentes promueve el diseño y desarrollo de aplicaciones en términos de entidades de software autónomas (agentes), situadas en un ambiente, y que pueden alcanzar con flexibilidad sus metas interactuando con otros agentes en términos de protocolos y lenguajes de alto nivel [7].

A. Organizaciones de Estandarización

Para controlar el desarrollo de la computación orientada a agentes se han formado organizaciones de estandarización, las más conocidas son MASIF (Mobile Agent System Interoperability Facilities Specification), KSE (Knowledge Sharing Effort) y FIPA (Foundation for Intelligent Physical Agents).

MASIF (Mobile Agent System Interoperability Facilities Specification)

Es un estándar de la OMG (Object Management Group), en el cual se definen los agentes móviles, éstos no se limitan al sistema donde iniciaron su ejecución, tienen la habilidad de transportarse de un sistema en una red a otra.

KSE (Knowledge Sharing Effort)

Es patrocinado por ARPA para el desarrollo de metodologías y software para compartir y reutilizar conocimiento. KSE ha definido estándares en la pragmática con KQML, en la sintaxis con KIF y en la semántica con el uso de ontologías.

En la parte pragmática, KQML (Knowledge Query and Manipulation Language), es un lenguaje y protocolo para intercambiar información y conocimiento que puede ser usado como un lenguaje en una aplicación que necesita interactuar con uno o mas sistemas inteligentes y compartir conocimiento para resolver un problema cooperativo.

En la sintaxis, KIF (Knowledge Interchange Format), es un lenguaje orientado a computador para el intercambio de conocimiento entre diferentes programas; es una semántica declarativa, lógicamente comprensiva que provee una forma para representar el conocimiento, reglas de razonamiento, objetos, funciones y relaciones. En la semántica, las ontologías en el concepto de compartir conocimiento, son una descripción formal de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes, se puede ver como un conjunto de definiciones de conceptos.

FIPA (Foundation for Intelligent Physical Agents)

Es una organización de estandarización de la Sociedad de Computo IEEE que promueve la tecnología basada en agentes y la interoperabilidad de sus estándares con otras tecnologías. FIPA definió el ACL(Agent Communication Language), el cual es un lenguaje para la comunicación entre agentes con una sintaxis muy similar a KQML.

B. Áreas de Trabajo

El auge en los SMA ha generado investigaciones en diferentes direcciones [22], las cuales se describen a continuación.

Modelamiento de Agentes

Se han planteado arquitecturas de agentes haciendo énfasis en diferentes características: agentes reactivos, lógicos, basados en deseo e intención, inteligentes, etc.

Arquitectura del SMA

Las arquitecturas de SMA planteadas se orientan a diferentes perspectivas, por ejemplo, se han planteado arquitecturas inspiradas en la sociedad, en modelos organizacionales y en modelos biológicos.

Metodologías de Desarrollo

Se han definido gran cantidad de metodologías y aún no se tiene un estándar para el análisis y diseño de SMA. Las metodologías tienen como propósito consolidar la experiencia adquirida en forma de metodologías [9]. La mayoría de las metodologías de agentes son producto de equipos de investigación académicos y se han probado en pequeños proyectos de desarrollo. Cada metodología maneja conceptos/ modelos diferentes porque aún no se tiene una estandarización [3]. Como esta ha sido una de las áreas de mayor estudio, en la siguiente sección se presenta un acercamiento mas profundo a las metodologías.

Técnicas de Notación y Diseño

Basándose en la técnica de notación orientada a objetos y componentes tradicional (UML), se propuso AUML como un estándar extendido de UML para los sistemas orientados a agentes. Esta notación no es totalmente aceptada porque las herramientas CASE no la soportan [4].

Infraestructuras del SMA.

Para soportar el desarrollo y ejecución de la infraestructura de los SMA se han desarrollado diversas herramientas que apoyan el proceso de análisis y diseño de SMA. Existen diferentes tipos de plataformas para el desarrollo de SMA [4], algunas sólo soportan la capa de transporte como JADE [1] y Fipa-Os. Otras herramientas realizan abstracciones de alto nivel, produciendo agentes poco robustos y eficientes, como: dMars, Jack, AgentBuilder, Agent Tool y Jam. Otra plataforma conocida es ParADE que soporta autonomía e interoperabilidad con las plataformas de la FIPA.

IV. METODOLOGÍAS DE DESARROLLO DE SMA

Teniendo en cuenta los conceptos/modelos de las metodologías se puede definir un agrupamiento según la principal perspectiva que tienen en cuenta en su desarrollo. Las metodologías que fueron formuladas teniendo presente el comportamiento inteligente de los agentes tienen una perspectiva de ingeniería del conocimiento y las metodologías que fueron formuladas teniendo en cuenta el proceso de desarrollo de software tienen una perspectiva de ingeniería de software. En la tabla 1 se muestran las principales metodologías de desarrollo de SMA clasificadas según las perspectivas definidas.

Tabla 1. Metodologías de Desarrollo de SMA

Perspectiva	Metodología
Ingeniería de Conocimiento	MAS-CommonKADS [13] PROMETHEUS [19]
Ingeniería de Software	ZEUS [6] GAIA [21] TROPOS [5] MASE [21] AAII [4] ROADMAP [19] CASSIOPIEDIA [4] INGENIAS [15] ALADDIN [14] RICA [17] ADELFE [2] PASSI [3]

A. Perspectiva de la Ingeniería del Conocimiento

Las metodologías clasificadas bajo la perspectiva de ingeniería de conocimiento proporcionan una buena base para modelar sistemas multiagente, ya que tratan el desarrollo de sistemas basados en conocimiento. Dado que los agentes tienen características cognitivas, estas metodologías pueden proporcionar las técnicas de modelado de la base de conocimiento de los agentes. Estas metodologías conciben un sistema basado en conocimiento como un sistema centralizado. Por tanto, no abordan los aspectos distribuidos o sociales de los agentes, la principal idea de estas metodologías es el control de la base de conocimientos por un agente central como se muestra en la figura 1.

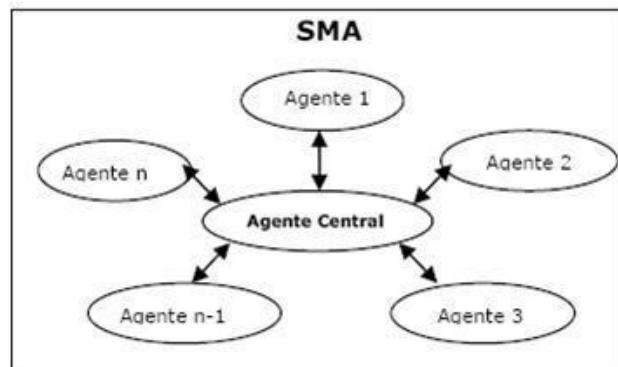


Figura 1. Sistema multiagente centralizado

Las metodologías clasificadas en esta perspectiva son MASCOMMONKADS y PROMETHEUS.

MAS-COMMONKADS [13] permite el desarrollo de un esquema detallado de sistemas inteligentes en su totalidad, cubriendo aspectos como comunicación, diseño y organización. Esta metodología está basada en CommonKADS [16], aportando una serie de modelos para desarrollar las fases de análisis y de diseño de sistemas multiagente. Esta metodología es clasificada en la perspectiva de ingeniería de conocimiento porque permite el modelamiento de agentes inteligentes por medio de un modelo de experiencia donde se describe el conocimiento necesario por los agentes para alcanzar sus objetivos.

PROMETHEUS [19] define un proceso detallado para la especificación, diseño, implementación y prueba de sistemas software orientados a agentes. Esta metodología fue clasificada en la perspectiva de ingeniería de conocimiento porque está orientada al diseño de agentes BDI. Las arquitecturas BDI se inspiran en un modelo cognitivo del ser humano teniendo en cuenta las creencias, los objetivos y los planes. Por estar inspirados en el modelo cognitivo del ser humano, estos tres modelos pueden ser vistos como la base de conocimiento de los agentes.

B. Perspectiva de la Ingeniería de Software

En la formulación de metodologías de desarrollo de SMA algunos investigadores se han enfocado en la adaptación de metodologías y conceptos de la ingeniería de software y de las metodologías orientadas a objetos, aunque no pueda hablarse de una definición formal de una metodología.

Estas metodologías presentan primordialmente tres vistas: una vista estática para describir la estructura de los agentes y sus relaciones; una vista dinámica para describir las interacciones

entre agentes; y una vista funcional para describir el flujo de datos entre las tareas de los agentes [3].

Las metodologías clasificadas en esta perspectiva son ZEUS, GAIA, TROPOS, MASE, AAI, ROADMAP, CASSIOPEIA, INGENIAS, ALADDIN, RICA, ADELFE y PASSI. A continuación se presenta una breve descripción de las metodologías.

ZEUS [6] propone un desarrollo en cuatro etapas: el análisis del dominio, el diseño de los agentes, la realización de los agentes y el soporte en tiempo de ejecución. ZEUS cuenta con una herramienta que soporta la realización de los agentes y el seguimiento en tiempo de ejecución.

GAIA [21] asume que el sistema es estático ya que las relaciones entre los agentes no cambian en el tiempo, por este motivo los agentes también son considerados estáticos porque sus habilidades y servicios no cambian [14].

TROPOS [5] propone una metodología de desarrollo de software fundamentada en los conceptos usados para modelar requisitos. En el marco de desarrollo se soportan los conceptos de actor, objetivo y dependencia; estos conceptos se utilizan para modelar requisitos, arquitectura y diseño.

MASE (Multi-agent systems Software Engineering) [21] considera los agentes como especializaciones de los objetos. La especialización consiste en que los agentes se coordinan unos con otros vía conversaciones y actúan proactivamente para alcanzar metas individuales y del sistema. En MaSE no se considera el razonamiento de los agentes, la organización de los agentes o la caracterización de su estado mental [9]. Esta metodología cuenta con una plataforma de desarrollo llamada Agent Tool.

AAII (Australian Artificial Intelligence Institute) [4] es una metodología para el análisis y desarrollo orientado a agentes basado en metodologías orientadas a objetos realizadas con algunos conceptos propios de agentes, como los modelos de comunicación.

ROADMAP [19] amplía la metodología de GAIA introduciendo casos de uso en el análisis de requisitos; modelos explícitos del ambiente y del conocimiento del agente; y un modelo de interacción basado en los diagramas de interacción de AUML.

CASSIOPEIA [4] propone tres pasos donde se identifican el comportamiento elemental aplicado en todo el sistema, las relaciones entre los comportamientos elementales y los comportamientos de organización del sistema que representa la manera en la cual los agentes se forman en grupos.

INGENIAS [15] profundiza en los elementos mostrados en el método de especificación y en el proceso de desarrollo. Esta metodología define un conjunto de meta-modelos (una descripción de alto nivel de los elementos que tiene un modelo) con los cuales se describe el sistema.

ALAADDIN [14] es una abstracción de un modelo de SMA genérico que se enfoca en la organización para resolver las desventajas de tecnologías centradas en agente.

RICA (Role / Interaction / Communicative Action) [17] es una metodología que integra aspectos relevantes de los lenguajes de comunicación del agente (ACL) y los modelos de organización. Esta metodología se basa en los conceptos de rol, interacción y comunicación.

ADELFE [02] es una metodología orientada a agentes y una herramienta que apoya notaciones de UML y AUML. El objetivo de la metodología es diseñar agentes con una actitud social enfocada a la cooperación.

PASSI (Process for Agent Societies Specification and Implementation) [03] plantea modelos y fases que abarcan la representación antropomorfa de los requisitos del sistema, punto de vista social, arquitectura de la solución, producción y reutilización de código y movilidad de los agentes.

V. DISEÑO DE UN SISTEMAS MULTIAGENTE PARA LA GESTIÓN DE INVENTARIOS

Como un acercamiento a la solución de la tarea de monitoreo de inventario, se plantea el diseño de un sistema multiagente distribuido que implemente un monitoreo constante en tiempo real y usando la red corporativa, del software instalado en las máquinas de los usuarios y los cambios realizados en el hardware.

El sistema multiagente es diseñado por medio de la metodología MAS-COMMONKADS [13], ya que diversos autores argumentan que esta es una de las metodologías más extensa y rigurosa con un grado de exactitud bastante alto [9]. Adicionalmente, en versiones posteriores del alcance de este trabajo, sería deseable adicionar características inteligentes a los agentes.

En general, las metodologías de la perspectiva de la ingeniería de conocimiento tienen la fortaleza de considerar las características cognitivas de los agentes, pero tiene la debilidad de no abordar aspectos distribuidos del SMA [9]. Estas

metodologías conciben un sistema basado en conocimiento como un sistema centralizado, por tanto, no abordan los aspectos distribuidos o sociales de los agentes, la principal idea de estas metodologías es el control de la base de conocimientos por un agente central.

La metodología seleccionada propone siete modelos para la definición del sistema: agente, tareas, experiencia, coordinación, comunicación, organización y diseño, como se presenta en la figura 2.

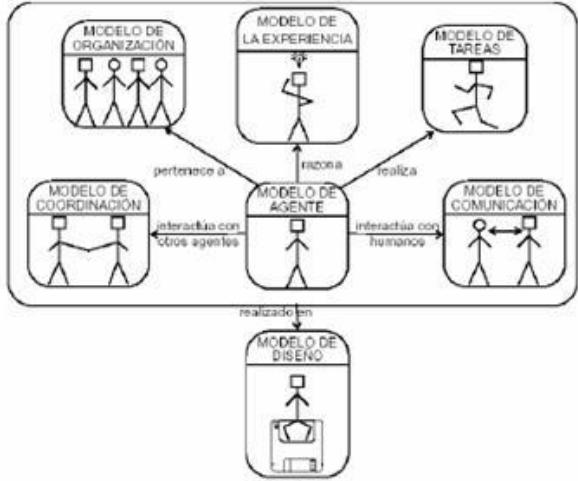


Figura 2. Modelos de MASCommonKADS, tomado de [13]

Al desarrollar los modelos propuestos por la metodología para diseñar un sistema de monitoreo distribuido de software y hardware, se obtiene la arquitectura presentada en al figura 3.

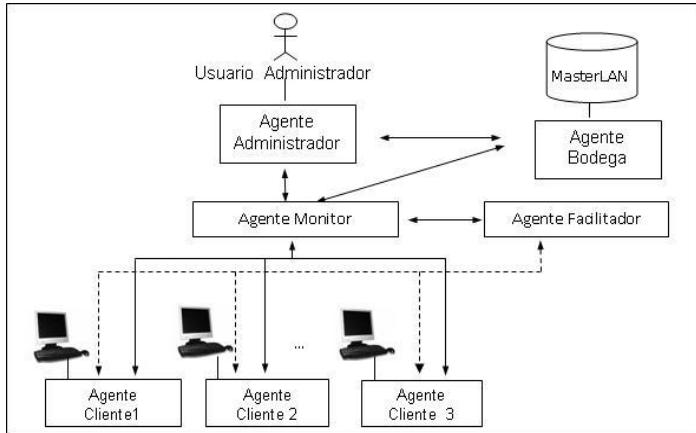


Figura 3. Arquitectura del SMA de Monitoreo de Inventario

En la arquitectura del SMA, se puede observar diferentes tipos de agentes, se tienen dos agentes de interfaz y tres agentes de tareas. Los agentes de interfaz son "Administrador" y "Bodega", los cuales se encargan de la comunicación con un usuario administrador del sistema y con la base de datos del sistema, respectivamente. Los agentes de tareas son "Monitor", "Facilitador" y "Cliente"; donde el primero se encarga de lanzar las solicitudes de monitoreo, el segundo se encarga de contactar a los agentes disponibles para realizar el monitoreo en los clientes y el tercero realiza las operaciones necesarias en los equipos clientes para capturar el informe de software y hardware.

El modelo de coordinación describe las interacciones que se presentan en el sistema, en el modelo los agentes se simbolizan con cabeza cuadrada y las conversaciones en las cuales participan son los óvalos. El modelo de coordinación construido se presenta en la figura 4.

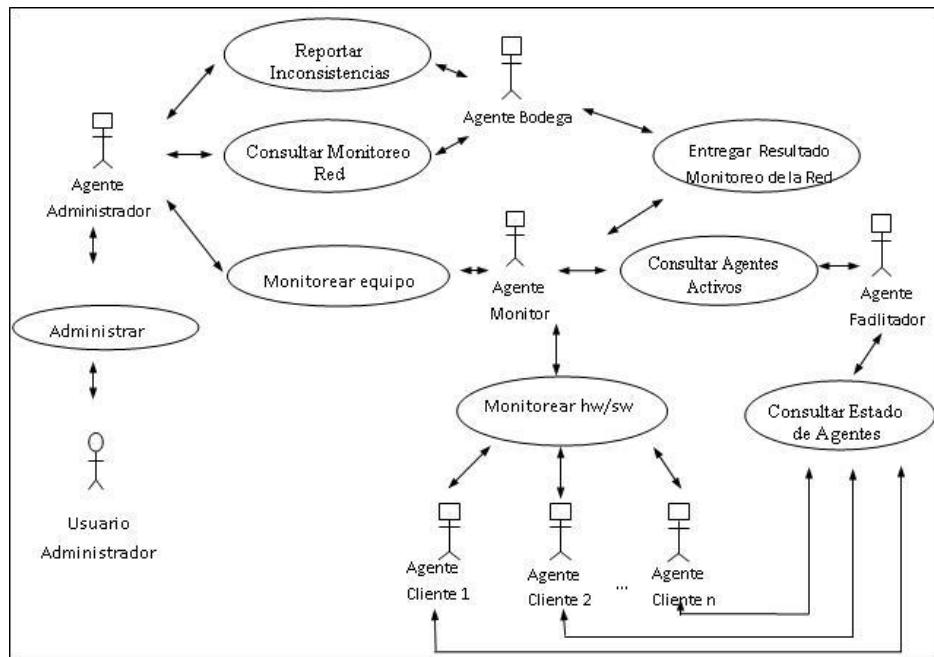


Figura 4. Arquitectura del SMA de Monitoreo de Inventario

VI. EVALUACION

Con el objetivo de apoyar la gestión de inventarios planteada en ITIL, se desarrolló un prototipo de la arquitectura de SMA planteada en la figura 3 para realizar un monitoreo distribuido y en tiempo real de la instalación de software y el robo de hardware en una red. Para monitorear los cambios en software y hardware se desarrolló una interfaz web que despliega toda la información requerida de las máquinas de la red.

El sistema desarrollado fue evaluado en una red windows pequeña con excelentes resultados de funcionamiento, ya que se logró cumplir con las siguientes prácticas de la gestión de inventarios:

- Ahorro de tiempo y esfuerzo para llevar a cabo los inventarios físicos. Mediante la implementación del sistema se reduce notablemente la cantidad de personas necesarias para realizar los inventarios de software y hardware.
- Facilidad de las labores de auditoría, proporcionando una visión general de los componentes de la red en tiempo real.
- Control de cambios y robos en piezas hardware.

VII. CONCLUSIONES

La gestión automática de inventarios permite a una organización tener una visión detallada de la configuración de hardware y software de las máquinas pertenecientes a la red en tiempo real, evitando los desplazamientos hasta el escritorio del usuario. Esta gestión está contemplada entre los procesos de gestión de configuración y gestión de activos de la fase de transición de servicios de ITIL.

En este trabajo se desarrolló un acercamiento a dicha gestión, logrando la automatización por medio de un sistema multiagente. Mediante la evaluación del sistema se puede verificar que se cumplen las buenas prácticas en la gestión de inventario propuestas en ITIL.

Como trabajo futuro se plantea que los agentes clientes sean móviles, para que viajen a través de la red hasta el cliente. También se plantea como trabajo futuro el monitoreo de información almacenada en memorias USB y otros dispositivos.

REFERENCIAS

- [1] Bellifemine F.; Poggi, A. y Rimassa, G., 2001. JADE: a FIPA2000 compliant agent development environment. En: Proceedings of the fifth international conference on Autonomous agents, ACM.
- [2] Bertron C.; Cossentino M.; Gleizes M.; Turci P. y Zambonelli F., 2004. A Study of Some Multiagent Metamodels. En: Agent-Oriented Software Engineering V: 5th International Workshop.
- [3] Bertron C.; Cossentino M. y Pavón J., 2005. Agent Oriented Software Engineering. En: The Knowledge Engineering Review, Vol. 20.
- [4] Bergenti F. y Poggi A., 2002. Agent-oriented software construction with UML. The Handbook of Software Engineering and Knowledge Engineering, Vol. 2, pp. 757-769.
- [5] Castro J. y Mylopoulos J., 2000. Tropos: A Framework for Requirements-Driven Software Development. En: Information Systems Engineering: State of the Artand Research Themes, Lecture Notes in Computer Science.
- [6] Collis J.; Ndumu D.; Nwana H. y Lee L., 1998. The Zeus Agent Building Tool-Kit. En: BT Technology Journal, Vol. 16(3), pp. 60-68.
- [7] Debenham J., 2002. Full lifecycle methodologies for agent-oriented systems the extended OPEN process framework. En: Proceedings of AgentOriented Information Systems.
- [8] Fipa 2004. FIPA: Agent Management Specification. Disponible electrónicamente en: <http://www.fipa.org>, consultado en septiembre de 2008.
- [9] Gomez J. J., 2003. Metodologías para el diseño de sistemas multiagente. En: Inteligencia Artificial, Revista Iberoamericana de I.A., Número 18, pp. 51-64.
- [10] Hewlett-Packard., 2008. Understanding inventory, configuration and IT asset management. Disponible en <http://h71028.www7.hp.com/ERC/downloads/4AA0-6093ENW.pdf>, consultado en septiembre de 2008.
- [11] Hochstein A.; Zarnekow R. y Brenner W., 2005. ITIL as common practice reference model for IT service management: formal assessment and implications for practice. En: IEEE International Conference.
- [12] Hochstein A.; Zarnekow R.; Brenner W., 2005. Evaluation of service-oriented IT management in practice; Services Systems and Services Management. En: Proceedings of ICSSSM '05.
- [13] Iglesias C. A.; Garijo M.; González J. C. y Velasco J. R., 1998. Analysis and design of multiagent systems using MAS-CommonKADS. En: Intelligent Agents IV LNAI, Vol. 1365, pp. 313-326.
- [14] Mao X. y YU E., 2004. Organizational and Social Concepts in Agent Oriented Software Engineering. En: AOSE 2004, agent-oriented software engineering V.
- [15] Pavón J.; Gómez J. y Fuentes R. 2003. INGENIAS Environment for MAS Generation. En: Workshop at CAEPIA-2003, X conferencia de la Asociación para la Inteligencia Artificial.
- [16] Schreiber G.; Akkermans H.; Anjewierden A.; de Hoog R.; Shadbolt N.; de Velde V y B Wielinga., 2000. Knowledge engineering and management, the CommonKADS methodology. Massachussets: MIT Press. 447 P.
- [17] Serrano J. M., 2004. Pragmática de los agentes software: análisis y diseño de lenguajes de comunicación artificiales. Universidad Rey Juan Carlos. PhD Thesis.
- [18] Schwartz L.; Ayachitula N.; Buco M.; Surendra M.; Ward C. y Weinberger S., 2007. Service Provider Considerations for IT Service Management. En: 10th IFIP/IEEE International Symposium.
- [19] Thomas J. y Sterling M. (2002). "Assembling Agent Oriented Software Engineering Methodologies from Features". En: Third International Workshop on Agent-Oriented Software Engineering, Bologna - Italy.
- [20] Wang H.; Yang B.; Liu L.; Ma Q.; Sun K. y Chen Y., 2007. Knowledge Enhanced IT Service Management. En: ICEBE 2007, IEEE International Conference.
- [21] Wooldridge M. J.; Jennings N. R. y Kinny D., 2000. The Gaia methodology for agent-oriented analysis and design. En: Autonomous Agents and Multi-Agent Systems, pp. 285-312.
- [22] Zambonelli F. y Omicini A., 2004. Challenges and research directions in agent-oriented software engineering. En: Autonomous Agents and Multi-Agent Systems, Vol. 9, pp. 253-283.

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas



Escuela de Ingeniería de Sistemas

Misión

La misión de la Escuela de Ingeniería de Sistemas es fomentar y apoyar la generación o la apropiación de conocimiento, la innovación y el desarrollo tecnológico en el área de ingeniería de sistemas e informática sobre una base científica, tecnológica, ética y humanística.



Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>

Visión

La formación integral de profesionales desde el punto de vista científico, tecnológico y social que les permita adoptar, aplicar e innovar conocimiento en el campo de los sistemas e informática en sus diferentes aspectos, aportando con su organización, estructuración, gestión, planeación, modelamiento, desarrollo, procesamiento, validación, transferencia y comunicación; para lograr un desempeño profesional, investigativo y académico que contribuya al desarrollo social, económico, científico y tecnológico del país.

