

MODELO MULTIAGENTE PARA EL DESARROLLO DE SISTEMAS DE GESTIÓN DEL CONOCIMIENTO



AUTORJ
UAN PABLO SOTO BARRERA
Licenciado en Ciencias Computacionales
Investigador del Grupo Alarcos
Universidad de Castilla La Mancha
jpsoto@proyectos.inf-cr.uclm.es
ESPAÑA

AUTOR
AURORA VIZCAÍNO BARCELÓ
Doctora Ingeniero en Informática
Investigador Grupo Alarcos
Universidad de Castilla La Mancha
aurora.vizcaino@uclm.es
ESPAÑA

AUTOR
JAVIER PORTILLO RODRÍGUEZ
Ingeniero en Informática
Investigador del Grupo Alarcos
Universidad de Castilla La Mancha
javier.portillo@alu.uclm.es
ESPAÑA

AUTOR
MARIO PIATTINI VELTHUIS
Doctor Ingeniero en Informática
Director del Grupo Alarcos
Universidad de Castilla La Mancha
mario.piattini@uclm.es
ESPAÑA

Fecha de Recepción del Artículo: 20 de Octubre de 2006 **Fecha de Aceptación del Artículo 4 de Diciembre de 2006**
Artículo Tipo 1

RESUMEN.

El desarrollo de Sistemas de Gestión del Conocimiento (SGC) es una tarea compleja que debe considerar: el proceso de generación del conocimiento, la identificación y manejo de los flujos de conocimiento, las técnicas de representación y recuperación de la información, así como los mecanismos utilizados para su distribución. En el desarrollo de estos sistemas es necesario encontrar un equilibrio entre cada uno de los aspectos anteriormente mencionados. Para alcanzar este objetivo se ha diseñado un modelo que permita a los ingenieros en ciencias computacionales el desarrollo de sistemas de gestión del conocimiento.

Nuestra propuesta, primeramente, define un modelo del ciclo de vida del conocimiento que, de acuerdo a la literatura y nuestra experiencia, considera cada una de las etapas que un sistema de gestión del conocimiento debe apoyar. Después, se describe la tecnología (agentes software) recomendada para dar soporte a las actividades que componen cada etapa del proceso de conocimiento. Además, se explica por qué consideramos que los agentes son una técnica adecuada para este fin y cómo ellos colaboran en la arquitectura propuesta intercambiando información. Por último se describe un prototipo para la gestión del conocimiento en el dominio de mantenimiento del software, este prototipo implementa los agentes inteligentes explicados en la arquitectura.

PALABRAS CLAVE

Gestión del Conocimiento
Sistemas de Gestión del Conocimiento
Sistemas Multiagentes
Agentes Software

ABSTRACT

Developing Knowledge Management Systems (KMS) is a complicated task since it is necessary to take into account how the knowledge is generated, how it can be distributed in order to reuse it and other aspects related to the knowledge flows. On

the other hand, many technical aspects should also be considered such as what knowledge representation or retrieval technique is going to be used.

To find a balance between both aspects is important if we want to develop a successful system. However, developers often focus on technical aspects giving less importance to knowledge issues. In order to avoid this, we have designed a model to help computer science engineers to develop these kinds of systems. In our proposal, firstly, we define a knowledge life cycle model that, according to literature and our experience, ponders all the stages that a knowledge management system should give support to. Later, we describe the technology (software agents) that we recommend to support the activities of each stage. The paper explains why we consider that software agents are suitable for this end and how they can work in order to reach their goals. Moreover, a prototype for the knowledge management in the software maintenance domain has been developed by using the agents described in the architecture.

KEYWORDS:

Knowledge Management
Knowledge Management Systems
Multiagents Systems
Software Agents

INTRODUCCIÓN

En la actualidad, el conocimiento constituye un recurso realmente competitivo de las empresas; debido a esto, la estructura y el entorno de producción de la organización juega un rol menos importante que el capital intelectual y la gestión del conocimiento [11]. Consecuentemente, una forma de calcular el rendimiento de una organización es determinar la forma en que ésta gestiona su conocimiento. Por lo tanto la manera de apoyar a las organizaciones en esta tarea es diseñar sistemas que faciliten la gestión del conocimiento organizacional. Estos sistemas son conocidos como Sistemas de Gestión del Conocimiento (SGC), definido por Alavi y Leidner [1], como un sistema de información diseñado concretamente para facilitar el proceso de de creación, almacenamiento, recuperación, transferencia y aplicación del conocimiento de una organización.

El desarrollo de un SGC no es una tarea fácil. En la literatura podemos encontrar varias propuestas para desarrollar este tipo de sistemas. Por ejemplo, el enfoque basado en el proceso/tarea que consiste en utilizar el conocimiento de los participantes de un proyecto, o el enfoque basado en el sistema (infraestructura) el cual consiste en construir un sistema base que permita capturar y distribuir el conocimiento en la organización [1]. Por otra parte, antes de desarrollar este tipo de sistemas se recomienda estudiar y entender cómo el conocimiento es transferido entre las personas en la vida real. Sin embargo, cuando los ingenieros desarrollan un SGC por lo común centran sus esfuerzos en la tecnología sin tener en cuenta los problemas fundamentales que el sistema debe soportar [10].

Existen varias técnicas para implementar un SGC, una de ellas

son los agentes inteligentes [32]. La tecnología de agentes software permite monitorizar y coordinar eventos o reuniones, así como la capacidad para diseminar la información [37]. Además, los agentes son proactivos es decir, poseen capacidad para tomar la iniciativa, sea mediante sugerencias o acciones anticipadas para alcanzar sus objetivos. Otra ventaja de esta tecnología es que los agentes aprenden de su propia experiencia. Por consiguiente, se espera que con el tiempo los sistemas basados en agentes lleguen a ser más eficientes puesto que aprenden de sus errores y éxitos [16].

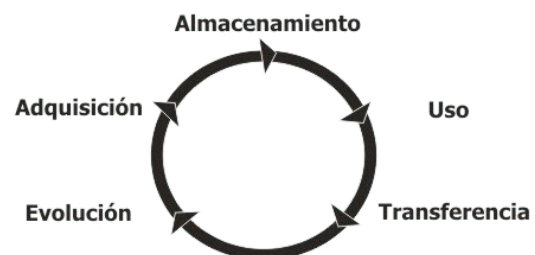
Debido a estas ventajas se han desarrollado diferentes arquitecturas multiagentes para gestionar el conocimiento [8]. Sin embargo, muchas de estas propuestas enfocan sus esfuerzos en un dominio en particular y son utilizadas bajo ciertas circunstancias. Otro problema, es que no toman en cuenta el ciclo del conocimiento. Debido a estas problemáticas, en este trabajo proponemos un modelo genérico para el desarrollo de SGC.

El resto del trabajo está organizado de la siguiente manera. En el apartado 1 se describe el modelo y los agentes software propuestos. En el apartado 2, se explica la estructura de cada agente y cómo han sido modelados usando la metodología de INGENIAS [23]. En el siguiente apartado describimos un prototipo que hace uso de los agentes propuestos en nuestro modelo. En el apartado 4 se exponen brevemente algunos de los trabajos relacionados con agentes. Finalmente, se presentan algunas conclusiones y trabajo futuro en el quinto apartado.

1. MODELO MULTIAGENTE PARA EL DESARROLLO DE SGC

Una implementación exitosa de un SGC debe ser capaz de llevar a cabo las funciones para crear, almacenar, recuperar, transferir y aplicar el conocimiento [13]. Teniendo en cuenta este hecho, y después de estudiar y analizar diferentes modelos y ciclos de vida del conocimiento [1, 4, 5, 19, 21, 28, 31, 34, 36], hemos decidido crear nuestro propio modelo donde se definen las etapas que un SGC debe apoyar (ver Figura 1). Las etapas de este ciclo son la adquisición, almacenamiento, uso, transferencia y evolución.

Figura 1. Modelo propuesto del ciclo de vida del conocimiento



Las primeras tres etapas se consideran en la mayoría de los ciclos de vida del conocimiento. Nosotros hemos agregado el uso (también considerado en algunos ciclos) y la transferencia, debido a que un SGC debe permitir a los usuarios utilizar el conocimiento de la organización (por ejemplo, recomendando la información que puede ser consultada o proponiendo soluciones a problemas). Además, el conocimiento debe actualizarse constantemente, de lo contrario nunca será consultado. Debido a esto, la etapa evolución ha sido agregada en este ciclo.

A continuación, describiremos cada una de las etapas del modelo de conocimiento propuesto. Al mismo tiempo se describen los agentes software que proponemos para dar soporte a cada una de las etapas del modelo.

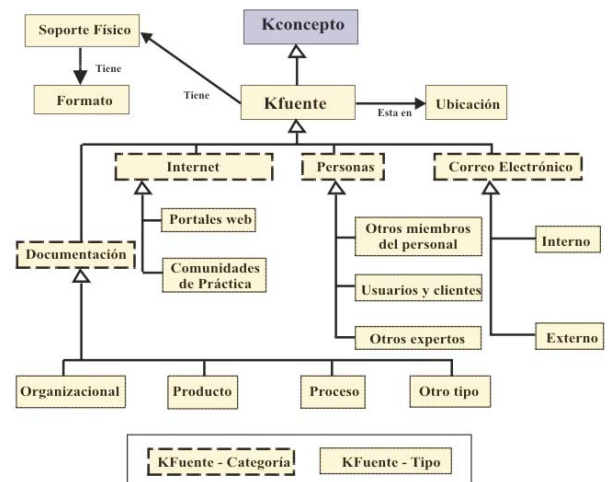
La Adquisición del Conocimiento es uno de los componentes claves en una arquitectura de gestión del conocimiento. Esta etapa incluye la elicitación, colección y análisis del conocimiento [25]. Durante la ejecución de este proceso es importante determinar dónde está localizado el conocimiento en la organización y cómo debe ser capturado. Para dar soporte a esta etapa proponemos utilizar un agente llamado Agente Capturador. El Agente Capturador es responsable de capturar la información (datos, modelos, experiencia, etc.) de las diferentes fuentes de conocimiento. Además, se encarga de llevar a cabo una monitorización constante para identificar la información y experiencias generadas entre el usuario y el sistema o herramientas groupware (correo electrónico, páginas web consultadas, chats, etc). Para lograr esto, el Agente Capturador puede utilizar diferentes técnicas, es decir, emplear herramientas y técnicas existentes que permiten la transformación de los datos corporativos en información [12], tales como:

- Sistemas demostrador o front-end (por ejemplo: DSS, EIS y OLAP).
- Sistemas back-end (por ejemplo: data warehouse, data mart y minería de datos) [8].

Los agentes también hacen uso de técnicas clásicas para obtener conocimiento, las cuales han sido utilizadas por expertos, tales como: entrevistas estructuradas, cuestionarios, árboles y redes de decisión, mapas conceptuales, etc. [25]. Además de estas técnicas los agentes pueden utilizar otras técnicas más sofisticadas como los analizadores de páginas web [3] usados para obtener información de Internet, clasificadores de documentos [22], administradores de listas de correos [18], redes neuronales, etc.

Una vez obtenido el conocimiento, éste debe ser clasificado de acuerdo al tipo y la fuente de conocimiento de la cual proviene. Para poder llevar a cabo esta tarea, el Agente Capturador hace uso de la ontología de las fuentes y tipos de conocimiento (ver Figura 2). Esta ontología ha sido diseñada en base a la ontología de Rodríguez et al. propuesta en [27].

Figura 2. Ontología de fuentes del conocimiento



La ontología utilizada por el Agente Capturador tiene cuatro categorías de fuentes de conocimiento. Estas son: documentación la cual se subdivide en documentación relacionada con la filosofía de la empresa, documentación que define los productos con los que la empresa trabaja, documentación que describe los procesos llevados a cabo dentro de la organización y la de otros tipo de documentos que no pueden ser clasificados en ninguno de los subcategorías descritas. Otra fuente importante en donde el Agente Capturador puede encontrar información es Internet, la cual se divide en otras subcategorías tales como portales, comunidades de práctica, etc. Sin embargo la principal fuente de conocimiento en las organizaciones son las personas, es por eso que hemos agregado dicha categoría. Dependiendo del tipo de compañía, las personas pueden clasificarse en empleados, socios, clientes, etc. Por último, la fuente del conocimiento representada en nuestra ontología es el correo electrónico el cual se divide en interno (correo electrónico enviado entre los empleados) y externo (correo electrónico enviado a otras personas externas de la organización).

La principal ventaja de este enfoque es que el Agente Capturador sería capaz de trabajar en cualquier dominio, ya que bastaría con adaptar las ontologías al nuevo dominio.

Formalización / Estructuración del conocimiento es la etapa que agrupa todas las actividades relacionadas con la organización, estructuración, representación y codificación del conocimiento con el fin de facilitar su reutilización [4]. Para poder llevar a cabo estas tareas hemos propuesto el Agente Constructor. Este agente es el encargado de dar el formato electrónico a las experiencias que son almacenadas en la base de conocimiento. La base de conocimientos es un componente de suma importancia para este tipo de sistemas ya que en ella son retenidos y hechos explícitos los conocimientos de los empleados expertos reduciendo de esta manera la dependencia de empleados claves. Para implementar esta actividad proponemos un Agente Constructor que compara la nueva información con el conocimiento previamente almacenado y decide si debe eliminar, añadir o combinar la información para generar nuevo

conocimiento. En este sentido el proceso de combinación del modelo SECI (propuesto en [20]) es realizado, produciendo nuevo conocimiento a partir de la combinación del conocimiento explícito con otro explícito.

Existen diferentes técnicas para almacenar el conocimiento, frecuentemente, la técnica utilizada se relaciona con el método de recuperación implementado. Es decir, si se decide utilizar la técnica de razonamiento basado en casos el conocimiento debería ser almacenado en forma de "casos".

Otras de las técnicas utilizadas son los objetos de conocimiento, frames, lógica de predicados o lógica difusa. En caso de utilizar ontologías para clasificar el conocimiento, es recomendable hacer uso de alguna de las metodologías existentes para su diseño. Ejemplos de estas metodologías son: Ontolingua [9] o REFSENO [30].

Uso del conocimiento es una de las principales etapas debido a que el conocimiento es útil cuando éste es usado o reutilizado. El principal enemigo de la reutilización del conocimiento es la ignorancia, es decir, en ocasiones los empresarios se quejan porque que los empleados no consultan las fuentes de conocimiento y no aprovechan el capital intelectual que la compañía tiene. Los SGC deben ofrecer la capacidad de buscar información; estos sistemas deben ser capaces de recomendar o sugerir conocimiento con el fin de facilitar las tareas de los empleados mediante la reutilización de las lecciones aprendidas. En nuestro modelo el agente encargado de esta actividad es el agente llamado Agente Buscador, el cual es el responsable de buscar el conocimiento requerido en la base de conocimiento. Actualmente existen diferentes tipos de técnicas para la recuperación de la información [6, 15]. Muchas de ellas se basan en el uso de la posición y frecuencia de palabras claves [17]. Otros autores, tales como [29] combinan varias técnicas como lo son: la minería de datos y el razonamiento basado en casos para el desarrollo de un sistema recomendador.

Transferencia del Conocimiento es la etapa que más atención ha recibido por parte de la comunidad investigadora del conocimiento [24]. Esta etapa se encarga de transferir tanto el conocimiento tácito y explícito. El conocimiento tácito puede transferirse si este fue previamente almacenado en medios compartidos, tales como: repositorios, memorias organizacionales, bases de datos, etc. La etapa de transferencia es implementada mediante el uso de mecanismos capaces de informar a los usuarios del nuevo conocimiento añadido en la base de conocimiento. Para esta etapa proponemos el Agente Diseminador, el cual debe identificar al grupo de personas o comunidades que generan y utilizan información similar, por ejemplo: en el dominio del software las personas que mantienen un mismo producto o aquellas que usan el mismo lenguaje de programación. Por lo tanto, este agente cubre la idea de comunidades de práctica en la cual cada persona comparte conocimiento y aprenden gracias a la comunicación con otros miembros de la comunidad [35]. Una apropiada integración de la gestión del conocimiento con las comunidades de práctica permite mejorar el desempeño de una organización [14]. La información diseminada puede ser de diferentes tipos; es

decir, información relacionada con la filosofía de la empresa o referente a los procesos. Finalmente, el Agente Diseminador necesita saber exactamente qué tipo de trabajo desempeña cada miembro de la organización, así como los flujos de conocimiento relacionados con sus actividades. Para llevar a cabo dicha tarea, el Agente Diseminador interactúa con un nuevo agente llamado Agente Personal el cual se encarga de identificar los perfiles de los usuarios que interactúan con el sistema. Haciendo una comparativa entre esta etapa con el modelo SECI podemos deducir que el Agente Diseminador cubre el proceso de socialización ya que pone en contacto a las personas que demandan la misma información con el fin de compartir sus experiencias, aumentando de esta manera el conocimiento tácito de los participantes.

Evolución del Conocimiento esta es la última etapa del ciclo de vida del conocimiento. Esta etapa consiste en monitorizar el conocimiento que por su naturaleza está en constante evolución. Para llevar a cabo esta actividad proponemos el Agente Mantenedor. El principal propósito de este agente consiste en mantener la base de conocimiento actualizada, por lo que aquella información que no es consultada es candidata a ser eliminada.

2. AGENCIAS MULTIAGENTES

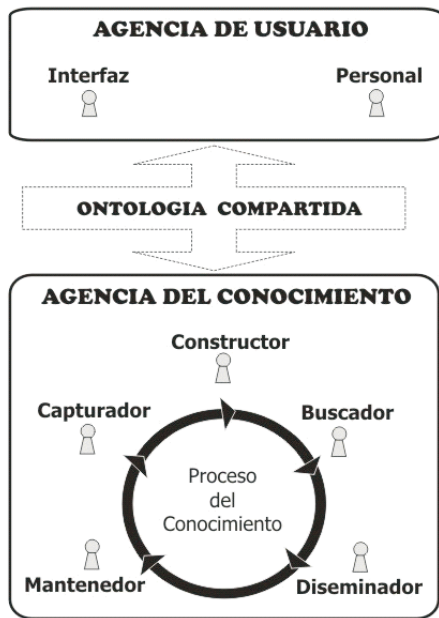
Una vez descrito el modelo y los agentes propuestos para la generación del conocimiento, es necesario describir la forma en que estos agentes trabajan y se comunican. Los agentes han sido agrupados en dos agencias:

- La Agencia del Conocimiento, la cual se encarga de dar soporte a los procesos de gestión de conocimiento. Esta agencia se compone de los siguientes agentes: Agente Capturador, Agente Constructor, Agente Buscador, Agente Diseminador y Agente Mantenedor.
- La Agencia de Usuario está formada por el Agente Personal y el Agente Interfaz. El Agente Personal se encarga de monitorizar las actividades llevadas a cabo por el usuario, con el fin de obtener sus preferencias y necesidades. El agente interfaz es el mediador entre los usuarios y los agentes. De esta manera, cuando un agente necesite enviar un mensaje al usuario, el agente deberá enviar el mensaje al agente interfaz para que este pueda mostrarlo al usuario.

Otro componente de nuestra arquitectura (ver Figura 3) es la Ontología Compartida, la cual permite conceptualizar el dominio de conocimiento. Esta ontología es utilizada para mantener una comunicación consistente entre ambas agencias.

Para llevar a cabo el análisis y diseño de los agentes involucrados hemos utilizado la metodología INGENIAS [23] la cual provee meta-modelos utilizados para definir sistemas multiagentes, así como las herramientas para generarlos. El uso de meta-modelos facilita enormemente el desarrollo de este tipo de sistemas, debido a que utiliza representaciones visuales que permiten cubrir los aspectos concretos del sistema.

Figura 3. Distribución de los agentes



A continuación, se ilustran dos diagramas que representan el meta-modelo de los agentes. Estos diagramas describen los roles y las tareas del Agente Capturador y el Agente Personal (un ejemplo de cada agencia). El resto de los diagramas son omitidos por cuestiones de espacio.

La Figura 4 muestra la meta principal del Agente Capturador que consiste en obtener la información que debe almacenarse. Su rol radica en “filtrar” puesto que decide qué información debe convertir en conocimiento, con el fin de que pueda ser utilizado en futuros proyectos. A continuación son descritas las tareas llevadas a cabo por el Agente Capturador.

Figura 4. Diagrama del Agente Capturador

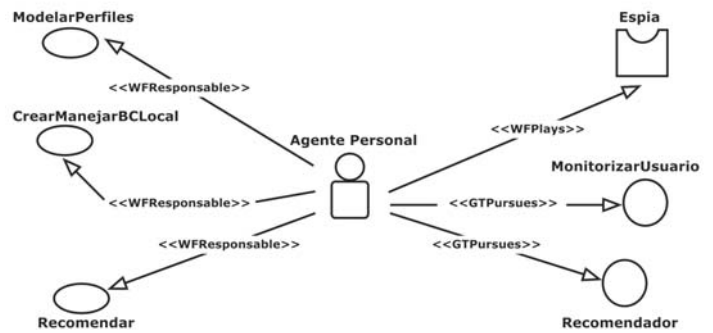


IdentificadorFI: Esta tarea radica en identificar las fuentes de conocimiento disponibles en el sistema.

CapturarInfo: Esta tarea consiste en capturar la información que será almacenada en la base de conocimiento.

EnviarConstructor: Una vez que la información ha sido analizada y es apropiada para su almacenamiento, el Agente Capturador envía la Información al Agente Constructor.

Figura 5. Diagrama del Agente Personal



La Figura 5 expone el diagrama del Agente Personal cuyo rol es llamado “espía” debido a que el agente debe monitorizar las actividades de los usuarios para obtener sus perfiles. Por lo tanto su metas consisten en monitorizar las tareas de los usuarios y la de recomendar información.

Para poder llevar a cabo estas metas se deben cumplir las siguientes tareas:

ModelarPerfiles: Esta tarea consiste en modelar el perfil del usuario. Para ello, el agente debe identificar las preferencias, actividades de los usuarios así como la información consultada.

CrearManejarBCLocal: Esta tarea consiste en crear y gestionar localmente una base de conocimiento que almacene el perfil del usuario.

Recomendar: Esta tarea consiste en recomendar el conocimiento o fuentes de conocimiento relevantes para el usuario.

3. PROTOTIPO DEL SISTEMA

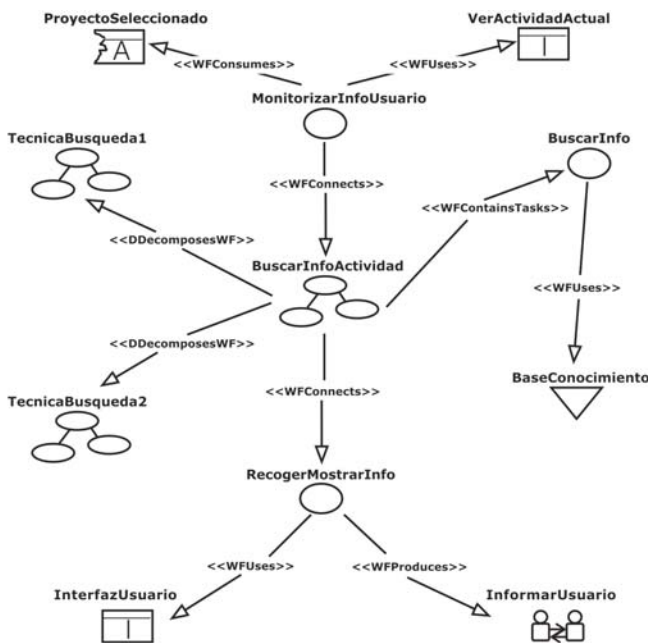
Para probar nuestro modelo hemos desarrollado un SGC para una empresa de mantenimiento de software. Hasta ahora, nuestro prototipo es capaz de recomendar las fuentes de información que los ingenieros del mantenimiento suelen consultar para resolver un determinado problema. Antes de construir el prototipo se hizo un estudio de los flujos de conocimiento existentes en las organizaciones enfocadas en el mantenimiento (ver [26]). Una manera de ilustrar la utilidad del prototipo es plantear un posible escenario.

Escenario

Un ingeniero de mantenimiento de software selecciona el proyecto en el cual va a trabajar. Después, el mantenedor empieza a trabajar en la actividad (por ejemplo en una petición de mantenimiento). Al mismo tiempo, el Agente Personal monitoriza los movimientos y actividades del mantenedor, registrando el nombre del proyecto y la actividad en que trabaja. Después, el Agente Personal envía un mensaje al Agente Buscador para obtener la información y conocimiento existente relacionado con la actividad que el mantenedor lleva a cabo. Dependiendo de la actividad el

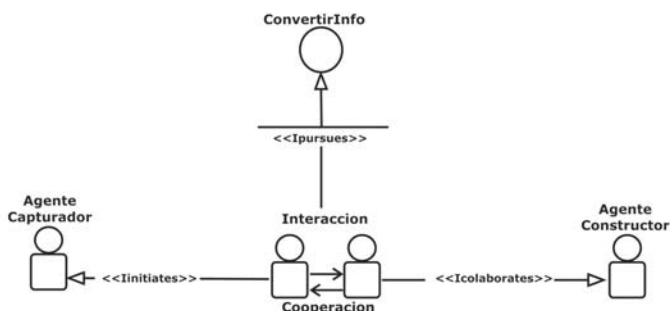
Agente Buscador utiliza una de la dos técnicas de recuperación de la información: posicionamiento y frecuencia de palabras claves (en el caso de que sea necesario proporcionar información relacionada con un tema en específico) o razonamiento basado en casos (en el caso que sea requerido proponer una solución a un determinado problema). Una vez que el Agente Buscador encuentra la información adecuada para resolver el problema, el agente envía la información al Agente Interfaz, el cual se encarga de informar al ingeniero del mantenimiento sobre la existencia del conocimiento necesario para realizar o facilitar la tarea que desempeña. La Figura 6 ilustra el diagrama de este escenario.

Figura 6. Diagrama del Escenario



Una vez que el ingeniero de mantenimiento finaliza su trabajo, el Agente Capturador verifica si un nuevo caso puede construirse (si el empleado encontró una nueva solución al problema) o si una nueva fuente de conocimiento fue utilizada. En ambos casos el Agente Capturador envía el nuevo conocimiento al Agente Constructor el cual se encarga de darle el formato adecuado y almacenarlo en la base de conocimiento o añadir nuevos conceptos a la ontología de fuentes de conocimiento.

Figura 7. Colaboración entre el Agente Capturador y el Agente Constructor



La colaboración entre el Agente Capturador y Constructor se representa en la Figura 7. Estos diagramas han sido generados siguiendo la metodología INGENIAS.

Algunos aspectos de la implementación

La plataforma de desarrollo utilizada es JADE (Java Agent Development Framework) debido a que cumple con el estándar FIPA y constituye una de las plataformas más utilizadas en el desarrollo de agentes. Además, JADE ha sido utilizada con gran éxito en el desarrollo de sistemas relacionados con la gestión del conocimiento [2, 7].

4. TRABAJO RELACIONADO

Los sistemas de gestión de conocimiento tradicionales han recibido un sin fin de críticas, debido a que los empleados son sobrecargados con trabajo extra, ya que estos SGC requieren de la captura de información por parte de los empleados. Una propuesta para evitar una carga extra a los empleados es añadir agentes software que lleven a cabo las tareas de captura y recuperación de la información.

Los beneficios de aplicar la tecnología de agentes en la gestión del conocimiento incluyen arquitecturas de sistemas distribuidos, facilidad para la interacción, administración de recursos, reactividad a cambios, interoperabilidad entre sistemas heterogéneos y toma de decisiones inteligentes.

La colección de tareas de gestión del conocimiento o aplicaciones a las que un agente puede asistir es muy variada, por ejemplo:

- Gestionar memorias organizacionales, un ejemplo puede ser el proyecto CoMMA (Corporate Memory Management through Agents) [7], que combina las tecnologías emergentes enfocadas en el aprovechamiento de las memorias organizacionales por parte de los usuarios.
- Dar soporte a actividades cooperativas. Por ejemplo en [33] los autores proponen una arquitectura multiagente para dar soporte a trabajo colaborativo.
- Recomendar. Por ejemplo en [29] se describe un sistema para llevar a cabo recomendaciones.
- Encontrar expertos. Algunos sistemas son utilizados para encontrar personas expertas que puedan dar asistencia a determinadas actividades dentro de la empresa.
- Manejador de listas de correo, o clasificador de la información [18].

Estos y otros sistemas a menudo han sido desarrollados sin considerar cómo el flujo de conocimiento fluye y las actividades que deben implementarse para favorecer este flujo de conocimiento en una empresa. Debido a eso, muchos SGC solamente dan soporte a una actividad, por ejemplo recomendación. Por otra parte, algunos de los previos SGC se centran en la tecnología, sin tomar en cuenta los problemas fundamentales que este tipo de sistemas deben afrontar [10].

5. CONCLUSIONES Y TRABAJO FUTURO

La aportación principal de este trabajo es el diseño de un modelo y una arquitectura multiagente para facilitar a los desarrolladores la implementación de SGC. Las ventajas de nuestro modelo son:

- El modelo está basado en un ciclo de vida del conocimiento. Por lo tanto, tratamos de evitar el descuido de otras arquitecturas que sólo se centran en la tecnología y olvidan los demás aspectos del conocimiento.
- El modelo proporciona asistencia a las diferentes etapas del conocimiento: creación, almacenamiento/recuperación, transferencia, aplicación y además evolución. Todas estas etapas (excepto la última), imprescindibles, según investigadores especializados en la evaluación de SGC, para que un SGC sea exitoso.
- El modelo propuesto hace uso de agentes software, los cuales evitan uno de los problemas de los SGC tales como sobrecargar con trabajo extra a los empleados en vez de ayudarlos en su trabajo diario.

Como trabajo futuro pretendemos hacer una comparativa entre los desarrollos de SGC basados en nuestra modelo con el desarrollo de otras arquitecturas. Por otra parte, estamos trabajando en ampliar la documentación de nuestro modelo con información detallada de las técnicas que pueden ser utilizadas para implementar cada uno de los agentes propuestos de acuerdo a las necesidades exigidas por las organizaciones.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto ENIGMAS (PIB-05-058), MECENAS (PB106-0024) y COMPETISOFT (506PI287) en España.

6. REFERENCIAS

1. Alavi, M., and Leidner, D. E.: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, Vol.25 (2001) 107-136.
2. Bergenti, F., Poggi, A., Rimassa, G.: Agent Architecture and Interaction Protocols for Corporate Memory Management Systems. *Proceedings of ECAI' 2000, Workshop on Knowledge Management and Organizational Memories*, (2000).
3. Camacho, D., Aler, R., Cuadrado, J.: Ruled-Based Parsing for Web Data Extraction. *Intelligent Agents for Data Mining and Information Retrieval*, M. Mohammadian, Ed.: Idea Group (2004) 65-87.
4. Davenport, T.H., Prusak, L.: *Working Knowledge: How Organizations Manage What They Know*. Project Management Institute. Harvard Business School Press, Boston, Massachusetts (1998) 199.
5. Dickinson, A.: WP3 - KM Framework. Enhancing Knowledge Management in Enterprises (ENKE) IST Project, IST-2000-29482, <http://www.ist-enke.com> (2000).
6. Frakes, B., Baeza-Yates, R.: *Information Retrieval Data Structures and Algorithms*. Englewood Cliffs, NJ: Prentice-

Hall, (1992).

7. Gandon, F.: A Multi-Agent Architecture for Distributed Corporate Memories. *Proceedings of the Third International Symposium from Agent Theory to Agent Implementation in European Meeting on Cybernetics and Systems Research* (2000) 623-628.
8. Giannella, C., Bhargava, R., Kargupta, H.: Multi-agent Systems and Distributed Data Mining. *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004*, Vol. 3191/2004. Springer-Verlag, Erfurt, Germany, (2004) 1-15.
9. Gruber, T.: A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, Vol. 5 (1993) 199-220.
10. Hahn, J., Subramani, M.: A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice. *21st International Conference on Information Systems (ICIS 2000)*, Brisbane, Australia (2000) 302-312.
11. Hansen, B., and Kautz, K.: Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organizations. *Proceedings of the EuroSPI (2004)* 126-137.
12. Houari, N., Homayoun, B.: Application of Intelligent Agent Technology for Knowledge Management Integration. *Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04)* (2004) 240-249.
13. Jennex, M., Olfman, L.: A Model of Knowledge Management Success. *International Journal of Knowledge Management*, Vol. 2 (2006) 51-68.
14. Lesser, E., Storck, J.: Communities of Practice and Organizational Performance. *IBM Systems Journal*, Vol. 40, Germany (2001) 831-841.
15. Liang, T., Huang, J.S.: A Framework for Applying Intelligent Agents to Support Electronic Trading. *Decision Support Systems*, Vol. 28 (2000) 305-317.
16. Maes, P.: Agents that reduce work and information overload. *Communications of the ACM*, Vol.37 (1994) 31-40.
17. Mohammadian, M., Jentzsch, R.: Computational Intelligence Techniques Driven Intelligent Agents for Web Data Mining and Information Retrieval. *Intelligent Agents for Data Mining and Information Retrieval*, Idea Group (2004).
18. Moreale, E., Watt, S.: An Agent-Based Approach to Mailing List Knowledge Management. *Proceedings of the Agent-Mediated Knowledge Management (AMKM)* (2003) 118-129.
19. Nissen, M.E.: An Extended Model of Knowledge-Flow Dynamics. *Communications of the Association for Information Systems*, Vol.8 (2002) 251-266.
20. Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, Vol. 5 (1994) 14-37.
21. Nonaka, I., Takeuchi, H.: *The Knowledge Creation Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press. Oxford University Press (1995) 304.
22. Novak, J., Wurst, M., Fleischmann, Strauss, W.: Discovering, Visualizing and Sharing Knowledge through Personalized Learning Knowledge Maps. *Proceedings of the Agents-Mediated Knowledge Management (AMKM)* (2003) 213-228.
23. Pavón, J., Gómez-Sanz, J.: Agent Oriented Software Engineering with INGENIAS. *Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*. LNCS 2691, Springer Verlag (2003) 394-403.

24. Peachey, T., Hall, D., Cegielski, C.: Knowledge Management and the Leading Information Systems Journals: An Analysis of Trends and Gaps in Published Research. *International Journal of Knowledge Management*, Vol. 1 (2005) 55-69.
25. Rhem, A.J.: *UML for Developing Knowledge Management Systems*. New York: Auerbach Publications (2006).
26. Rodriguez, O., Martinez-Garcia, Ana., Vizcaino, A., Favela, J., Piattini, M.: Identifying Knowledge Flows in Communities of Practice. In: In Coakes, Elayne and Steve Cleark (ed.): *Encyclopedia of Communities of Practice in Information and Knowledge Management*. IDEA Group (2005) 210-217.
27. Rodriguez, O., Martínez, A., Favela, J., Vizcaino, A., Piattini, M.: Understanding and Supporting Knowledge Flows in a Community of Software Developers. *Proceedings of the X International Workshop on Groupware (CRIGW)*, LNCS 3198, Springer, San Carlos, Costa Rica (2004) 52-66.
28. Rus, I., Lindvall, M.: Knowledge Management in Software Engineering. *IEEE Software*, Vol. 19 (2002) 26-38.
29. Sung, K.J.: Customized Recommendation Mechanism Based on Web Data Mining and Case Based Reasoning. *Intelligent Agents for Data Mining and Information Retrieval* (2004).
30. Tautz, C., Von Wangenheim, C.: REFSENO: A Representation Formalism for Software Engineering Ontologies. Fraunhofer IESE-Report 015.98/E, version 1.1 October. (1998).
31. Tiwana, A.: *The Knowledge Management Toolkit: Practical Techniques for Building Knowledge Management Systems*. Prentice-Hall, USA (2000) 608.
32. van-Elst, L., Dignum, V., Abecker, A.: Agent-Mediated Knowledge Management. *International Symposium AMKM 2003*. Springer, Stanford, CA, USA (2003) 1-30
33. Wang, A., Reidar C., Chunnian, L.: A Multi-Agent Architecture for Cooperative Software Engineering. *Proceedings of the Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE'99)*, Germany (1999) 162-169
34. Ward, J., Aurum, A.: Knowledge Management in Software Engineering - Describing the Process. *Proceedings of Australian Software Engineering Conference, ASWEC'04*, Melbourne, Australia (2004)
35. Wenger, E.: *Communities of Practice: Learning Meaning, and Identity*. Cambridge University Press, Cambridge U.K. (1998).
36. Wiig, K.M.: Knowledge Management: Where Did it Come from and Where Will it Go?: *Expert Systems with Applications*, Vol. 13 (1997) 1-14.
37. Wooldridge, M., Jennings, N.R.: Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, Vol. 10 (1995) 115-152.