

Memoria Organizacional Basada en Casos para dar Soporte al Proceso de Aseguramiento de Calidad

María de los A. Martín, Luis Olsina

GIDIS_Web, Departamento de Informática, UNLPam
Calle 9 y 110, (6360) General Pico, La Pampa, Argentina
E-mail martinma.olsinal@ing.unlpam.edu.ar

Resumen. En los últimos años han surgido numerosas propuestas de modelos y herramientas de gestión del conocimiento que tienen como objetivo administrar y almacenar el conocimiento organizacional. Sin embargo, la mayoría de ellas almacena el conocimiento en forma no estructurada o semi-estructurada, dificultando el procesamiento semántico y automático de dicho conocimiento. En este trabajo se propone un marco conceptual y el diseño de una memoria organizacional basada en casos de manera que pueda ser utilizada para aprender, resolver problemas y como apoyo en la toma de decisiones. El modelo propuesto está basado en una ontología y usa tecnologías de Web Semántica, lo que facilita el reuso, la interoperabilidad y el procesamiento automático por parte de agentes. Para ilustrar la utilidad de este modelo, se muestra un caso práctico: un sistema de recomendación como apoyo al área de aseguramiento de calidad en Ingeniería de Software.

Abstract. With the aim to manage and retrieve the organizational knowledge, in the last years numerous proposals of models and tools for knowledge management have arisen. However, most of them store the knowledge in a not structured or semi-structured way, hindering the semantic and automatic processing of this knowledge. In this work we introduce a conceptual framework as well as the design of an organizational memory based on cases, so that it can be used to learn, to solve problems, and as support in decision making as well. The proposed model is based on an ontology that allows using semantic Web technologies, which in turn facilitates the reuse, the interoperability and the automatic processing by agents. In order to illustrate the utility of this model, a practical case is shown: A recommendation system as support to the quality assurance area in Software Engineering.

Palabras Clave. Memoria Organizacional, Administración del Conocimiento, Razonamiento Basado en Casos, Web Semántica, Aseguramiento de Calidad.

1 Introducción

Los sistemas de administración del conocimiento (KMS) tienen como objetivo administrar y almacenar el conocimiento organizacional, de manera que después pueda ser utilizado para aprender, resolver problemas y como apoyo en la toma de decisiones [Con96, Dog93]. Además facilita el ahorro de tiempo y dinero, permitiendo conocer los errores y aciertos del pasado para mejorar el presente sin repetir esos errores y tomando ventaja de las “lecciones aprendidas”.

La mayoría de los KMS que existen en la actualidad capturan el conocimiento en repositorios de documentos como manuales, memorándums, y en sistemas informáticos de archivos de texto, y su transferencia se hace por medio de reuniones de trabajo, cursos de capacitación y lecturas individuales de manuales y guías. Esta forma tradicional de almacenar y transferir este conocimiento, ocasiona grandes pérdidas de tiempo y alta inversión en recursos humanos, ya que no contemplan mecanismos potentes de procesamiento semántico y automático de dicho conocimiento.

Una forma de resolver este problema es almacenar el conocimiento en forma estructurada en lo que se denomina “memoria organizacional”. La memoria organizacional va a ser, por lo tanto, el almacenamiento de todo el conocimiento formal e informal [Con96] presente en la organización.

En este paper se presenta una ontología de una memoria organizacional basada en casos, y el diseño de un sistema de administración de la memoria organizacional basado en dicha ontología. La estructuración del conocimiento en casos, le imprime mayor capacidad de procesamiento y facilita su captura, recuperación, transferencia y reuso. Además con el fin de agregar interoperabilidad, generalidad y potencia de procesamiento, la arquitectura del sistema propuesto usa tecnologías de Web Semántica.

El resto del artículo se organiza de la siguiente manera. En la sección 2 se describen brevemente los principales conceptos de administración del conocimiento y memoria organizacional, útiles para una mejor comprensión de este trabajo. En la sección 3 se describe la representación del conocimiento usando casos. En la sección 4 se introduce la ontología de memoria organizacional basada en casos. En la sección 5 se muestra la arquitectura propuesta para el sistema de administración de la memoria organizacional basada en casos (MOBC). Para ilustrar la utilidad de la memoria organizacional, en la sección 6 se desarrolla un caso práctico, un sistema de recomendación como apoyo al área de aseguramiento de calidad en Ingeniería de Software. Finalmente en la sección 7 se exponen las conclusiones y líneas de avance.

2. Sistemas de Administración del Conocimiento y Memoria Organizacional

Una organización que aprende es una compañía que constantemente construye estructuras y estrategias tales que incrementen y maximicen el conocimiento organizacional [Dog93]. El conocimiento es una ventaja clave de la organización. La memoria organizacional extiende esta ventaja por medio de la captura, reuso organización y diseminación y del conocimiento creado por sus empleados [Mal96].

Las compañías dedicadas al software no son una excepción, son empresas donde las personas continuamente expanden su capacidad para crear los resultados que ellos realmente desean, donde nuevos patrones de pensamiento se nutren, y donde las personas continuamente aprenden de sus experiencias. Y la memoria organizacional no es sólo un facilitador para la acumulación y mantenimiento de dicho conocimiento, sino también para compartirlo.

Para que este conocimiento sea útil en una organización, el KMS debe incluir un sistema de recuperación del mismo de suficiente calidad, que sirva de apoyo a la toma de decisiones dentro de la organización, basado en las experiencias pasadas almacenadas en el sistema, con el fin de no repetir errores anteriores. Además, y teniendo en cuenta que el conocimiento deberá ser almacenado en forma estructurada aparece el concepto de “memoria organizacional”. La memoria organizacional va a ser, por lo tanto, el almacenamiento de todo el conocimiento formal e informal [Con96] presente en la organización. El primero surge de los documentos, guías de buenas prácticas, manuales y libros que apoyan a los miembros de la organización a tener un buen desempeño. El conocimiento informal, en cambio, surge de la experiencia y el proceso de creación de los miembros. Aquí se incluyen las ideas, hechos, significados, presunciones, preguntas, puntos de vista y decisiones.

3. Representación del Conocimiento usando Casos

El contenido de una memoria organizacional va desde lo documentado (reportes de proyectos, modelos, planificación, manuales de procedimientos, etc.) hasta lo no documentado (experiencias, formas de pensar, anécdotas) pero que forman el acervo cultural, conocimientos y experiencias de los miembros de la empresa. Si bien el conocimiento documentado es un capital valioso dentro de una organización, es de vital importancia tener en la memoria organizacional aquel conocimiento que permita contestar preguntas como: ¿Porqué se hizo esto de cierta manera?, ¿Este problema ha sido resuelto antes?, ¿Qué aprendimos la última vez que sucedió ese problema?. Una de las maneras de lograrlo, es guardar estas experiencias por medio de casos.

3.1 Casos

Un caso es una pieza contextualizada de conocimiento que representa una experiencia. Contiene la lección pasada que es el contenido del caso y el contexto en el cual la lección puede ser utilizada [Kol93]. Típicamente un caso comprende:

- El problema que describe el estado del mundo cuando ocurrió el caso.
- La solución que describe cómo se resuelve el problema, y/o
- El resultado que describe el resultado obtenido como consecuencia de la solución del problema.

CASO: Casos de testing de un módulo con dos variables de entrada.
PROBLEMA: Diseñar Casos de testing para un módulo con dos variables de entradas X e Y .
SOLUCIÓN: Los casos de prueba diseñados fueron 9: 1) X=valor válido ; Y= valor válido 2) X=valor válido ; Y= valor en el límite de validez 3) X=valor válido ; Y= valor inválido 4) X=valor en el límite de validez ; Y= valor válido 5) X=valor en el límite de validez; Y= valor en el límite de validez 6) X=valor en el límite de validez; Y= valor inválido 7) X=valor inválido ; Y= valor válido 8) X=valor inválido ; Y= valor en el límite de validez 9) X=valor inválido : Y= valor inválido
RESULTADO: Todos los defectos existentes fueron detectados.

Figura 1. Ejemplo de Caso para el dominio de Testing en Ingeniería de Software.

De acuerdo a lo anterior, un caso se puede definir como la descripción detallada de una experiencia del pasado sobre una situación particular, formada por la descripción del problema, la solución tomada para resolver el problema y el resultado obtenido después de aplicada la solución. En la figura 1 se muestra un ejemplo de caso relacionado al área de desarrollo de software.

Los casos que incluyen un problema y solución, pueden usarse para derivar soluciones a nuevos problemas, basándose en experiencias similares del pasado, tal como lo define el método de razonamiento basado en casos (CBR) [Aam94]. Si además tienen una descripción del resultado, pueden usarse para evaluar propuestas de solución y anticipar problemas potenciales antes de que ellos ocurran [Kol93].

Tradicionalmente, hay varios tipos de métodos para representar casos, que van desde representaciones no estructuradas a totalmente formales y automáticamente procesables [Che03]; estos métodos son:

- **Método textual:** Un caso se representa en un formato de texto libre, o por una lista de preguntas y respuestas. Por ejemplo, el caso ilustrado en la figura 1 está representado de esta forma. Este método es el más potente en cuanto a la capacidad de representación y el más simple, pero necesita la lectura humana para su interpretación.
- **Representación a través de atributos:** Un caso se representa con pares (atributo, valor), por ejemplo, en el caso de la figura 1, el problema podría ser representado por los pares: (CantidadVariables, 2), (ValorMinimoX,10), etc. Cada atributo tiene un tipo y rango de valores asociado. Esta representación, si bien puede ser procesada automáticamente no guarda información estructural y relacional, por lo tanto falla al describir objetos complejos, por ejemplo, un módulo de código, una métrica, etc, son objetos estructurados que no pueden ser representados con un par (atributo,valor).
- **Representación estructurada:** Consiste en aplicar, por ejemplo, técnicas orientadas a objetos para representar casos. Este tipo de representación es la más adecuada para dominios complejos, en el cual los casos involucran variables estructuradas. Los casos son representados como una colección de objetos, cada uno de los cuales es descrito por un conjunto de atributos, que a su vez pueden ser simples o estructurados.

4. Ontología de Memoria Organizacional basada en Casos

Últimamente, se ha puesto mucha atención al desarrollo de sistemas que permiten compartir el conocimiento publicándolo en la Web. Sin embargo, su tamaño, complejidad y heterogeneidad hace que la distribución del conocimiento, no sea una tarea fácil. Afortunadamente, el W3C (World Wide Web Consortium), ha impulsado el desarrollo de la “Web Semántica”, con un conjunto de tecnologías que enriquecen la información disponible en Internet, proveyéndola de su semántica, y recomienda un nuevo paradigma para compartir e intercambiar conocimiento, basado en ontologías [Fen03].

Los sistemas integrados de aplicaciones Web y aplicaciones empresariales actuales, carecen de estándares y descripciones formales, lo cual dificulta la compartición y reuso de información y la cooperación entre individuos, organizaciones y aplicaciones. Las ontologías pueden resolver potencialmente este problema facilitando la compartición y reuso del conocimiento a través de la definición formal de la semántica de los conceptos del mundo real (ver como ejemplo, la ontología de métricas e indicadores [Mar03], [Ols04] desarrollada para un marco de medición y evaluación [Ols05], con base en Web Semántica [Mol04]). En esta sección se presenta una ontología de memoria organizacional basada en casos, y en la siguiente sección el diseño de un marco de memoria organizacional basada en dicha ontología.

4.1 Conceptualización

En esta sección describimos los principales conceptos de la ontología de memoria organizacional basada en casos, que se ilustran en el diagrama UML de la figura 2. Una memoria organizacional basada en casos, es un repositorio que almacena el conocimiento adquirido en experiencias pasadas, como son lecciones aprendidas, buenas prácticas, heurísticas, etc, es decir casos. Para una mejor organización y búsqueda de dichas experiencias, la memoria organizacional se compone de varias bases de conocimientos basada en casos, que agrupa los casos por tipo de

conocimiento. El conocimiento almacenado es referido a un determinado dominio de aplicación, por lo tanto la memoria organizacional está relacionada con una ontología de dominio.

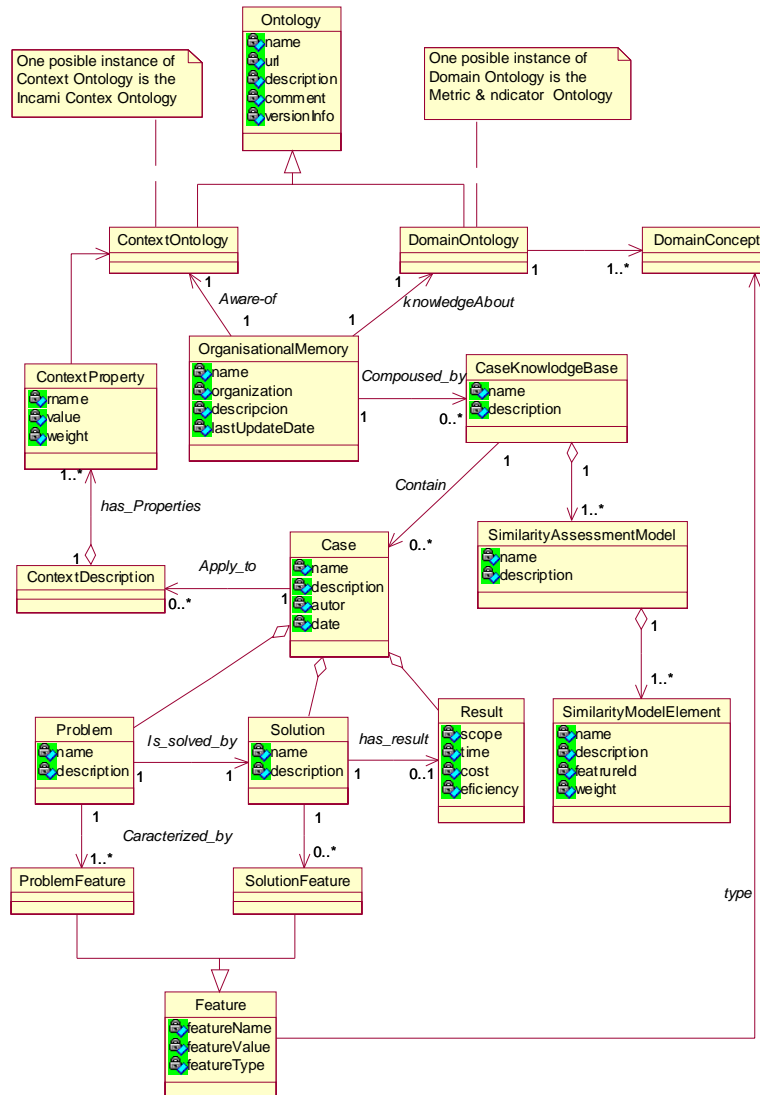


Figura 2. Modelo conceptual de la ontología de memoria organizacional basada en casos

Según la definición de Kolodner [Kol93] “Un caso es una pieza contextualizada de conocimiento que representa una experiencia”, por lo tanto es fundamental en toda memoria organizacional guardar la información del contexto donde ocurre cada caso. La representación del conocimiento a través de casos, facilita el reuso del conocimiento adquirido en situaciones de problemas similares para ser aplicado a un nuevo problema [Aam94].

Como el objetivo de nuestra ontología es que sirva como base para el intercambio de conocimiento organizacional pero también (y no menos importante) como base para un sistema de recomendación con potencia semántica, donde el significado de los conceptos debe estar formalmente especificado para que pueda ser procesado por computadoras, es que optamos por el método de representación estructurada para elaborar el modelo conceptual de los casos. En una definición formal un caso es un par ordenado <P, S>, donde:

P es el espacio del problema;

S es el espacio de la solución;

Como un mismo problema, puede tener varias soluciones, no todas con el mismo resultado, para una mejor elección de una solución en la toma de decisiones, se necesita guardar también el resultado obtenido en la aplicación de cada solución. Por lo tanto, un caso se compone de un problema, una solución y un resultado (ver el modelo UML de la figura 2).

Hay una descripción general de problemas P (x_1, x_2, \dots, x_n) y cada problema individual es una instancia P(a_1, a_2, \dots, a_n) y también hay una descripción general de solución S (y_1, y_2, \dots, y_m) donde toda solución individual S (b_1, b_2, \dots, b_m) es una instancia de la descripción general de solución. Los x_i son variables de características del problema (problem feature) y los y_i son variables de características de la solución (solution feature). El proceso de razonamiento basado en casos consiste en asignar valores a las variables de características del problema (caracterizar el problema), y encontrar los valores adecuados para las instancias de la solución, a través de criterios de evaluación de similitud de casos.

Para cada tipo de conocimiento se debe especificar un modelo de similitud, que le permita al razonador basado en casos evaluar la similitud entre ellos. El resultado de un caso debe permitirnos evaluar la conveniencia de aplicar una solución u otra, y representa una medida de la calidad de la solución, por lo tanto, forma parte del resultado de una solución, su alcance, tiempo, costo y eficacia.

Para que una memoria organizacional pueda ser implementada en la Web semántica, y pueda ser procesada automáticamente, necesita tener asociada una ontología de dominio (ver como ejemplo [Ols04]) que proporcione la terminología y tipos de datos necesarios para definir los casos en forma no ambigua, de manera de asegurar un intercambio de conocimiento eficaz y el razonamiento basado en casos en forma distribuida. Esta ontología de dominio proporciona la terminología (donde cada concepto está definido como una clase OWL o tecnología similar de la web semántica) que proporciona los tipos de las variables que caracterizan al problema y a la solución, es decir proporciona los feature types. Para un caso como el del ejemplo de la figura 1, se debiera proveer de una ontología para el dominio de testing que proporcione términos como: caso de prueba, modulo, parámetro, precondition, post-condición, etc., ya existen ontologías desarrolladas para muchos de los tópicos de Ingeniería de Software, como citamos previamente.

Con respecto a la representación de información de contexto ocurre algo similar, según la definición de Dey: "Contexto es cualquier información que puede ser usada para caracterizar la situación de una entidad, siendo una entidad una persona, lugar, u objeto que se considera relevante en la interacción entre un usuario y una aplicación, incluyendo también a ellos mismos, usuario y aplicación" [Dey01]. En términos generales, un sistema es consciente del contexto, si lo usa para proporcionar información y/o servicios, donde la relevancia de dicha información depende del contexto actual. De igual modo, en las memorias organizacionales basada en casos, la relevancia de la información de experiencias similares, está muy ligada al contexto en que es válido aplicar dicha experiencia. En este caso, y volviendo a la definición de Dey, la entidad relevante en la interacción entre el usuario y la aplicación es el caso. Debemos definir cuál es la información que puede ser usada para caracterizar la situación de un caso, es decir su contexto de aplicación. Los parámetros para caracterizar dicho contexto (context feature), están definidos en la ontología de contexto, por lo tanto, una memoria organizacional, tiene asociada una ontología de contexto, que permitirá evaluar el contexto de cada caso que almacena. En la tabla 1 se ilustra la definición de los algunos conceptos de la ontología.

Tabla 1. Ontología de Memoria Organizacional Basada en Casos: Glosario de Conceptos.

Nombre	Descripción
Ontología	Especificación formal de una conceptualización compartida [Gru95].
Ontología de Dominio	Ontología que expresa conceptualizaciones que son específicas a un dominio particular [Cor01]. Ontoweb Consorcio.
Ontología de Contexto	Ontología de dominio que especifica los parámetros de un contexto.
Memoria Organizacional	El modo en que una compañía almacena y gestiona el seguimiento de su conocimiento, los procesos de negocio y habilidades de sus empleados. (Center for Coordination Science. Massachusetts Institute of Technology. MIT. http://ccs.mit.edu/21c/iokey.html)
Caso	Pieza contextualizada de conocimiento que representa una experiencia. [Kol93].
Contexto	Cualquier información que pueda ser usada para caracterizar la situación de una entidad. [Dey01].
Problem	Un estado de dificultad que necesita ser resuelta [Wordnet]
Solution	Una sentencia que resuelve un problema o explica como resolverlo. [Wordnet]
Result	Consecuencia obtenida por la aplicación de una solución a un problema. [Wordnet]

5. Arquitectura de la Memoria Organizacional Basada en Casos

La Arquitectura de Memoria Organizacional Basada en Casos (MOBC) que se presenta en esta sección, es una arquitectura genérica, basada en la Web y pensada para ser desarrollada con tecnologías de Web Semántica. Permite el almacenamiento persistente de casos y su posterior consulta en línea. En la figura 3 mostramos una vista general de esta arquitectura, y cómo se puede usar como componente principal de un Sistema de Administración del Conocimiento. Para el diseño de la arquitectura MOBC, se ha elegido un estilo arquitectónico multi-nivel o de n-capas. Este estilo arquitectónico proporciona a la aplicación una estructura clara y comprensible, siendo más fácilmente modificable y escalable. Las capas de la arquitectura son:

La capa de interfaz. Proporciona acceso personalizado al sistema de administración del conocimiento, para los usuarios humanos, ofreciendo todas las funcionalidades de un KMS a saber: Creación y captura del conocimiento, búsqueda y acceso al conocimiento, diseminación del conocimiento, uso y mantenimiento del conocimiento. Para implementar la funcionalidad, esta capa redirigirá las peticiones a la capa lógica.

La capa lógica. Implementa la funcionalidad que el sistema debe proporcionar a los usuarios, agentes o herramientas. Esta funcionalidad se implementa mediante una librería de API's (Application Programming Interface) que ofrece métodos específicos a sus clientes y los traduce en invocaciones a distintos métodos de los DBMS apropiados. Dos ventajas importantes de contar con esta capa separada son, por un lado la posibilidad de implementar la memoria organizacional arriba de una gran variedad de repositorios en distintos DBMS, sin tener que cambiar ninguna componente; y por otro, que la funcionalidad provista por la arquitectura, puede ser accedida tanto por agentes y herramientas como por aplicaciones. Las funcionalidades de esta capa se agrupan en tres módulos:

- **El módulo de consultas:** Este módulo evalúa tanto consultas sobre los datos semánticos, recuperando los esquemas RDF (Resource Description Framework), como consultas de casos específicos, o casos similares (incluye un motor de razonamiento basado en casos).
- **El módulo de administración:** Este módulo permite la actualización incremental de los esquemas RDF, como también su eliminación (se debe tener permisos especiales para esta funcionalidad).

- **El módulo de importación:** Este módulo permite la importación de casos desde otras memorias organizacionales distribuidas.

La capa de acceso a datos. Se integra de un conjunto de componentes software que independizan y facilitan a las capas superiores el acceso a los datos del repositorio almacenados en la capa de persistencia.

La capa de persistencia. Almacena de manera persistente, tanto la información de casos, como las ontologías.

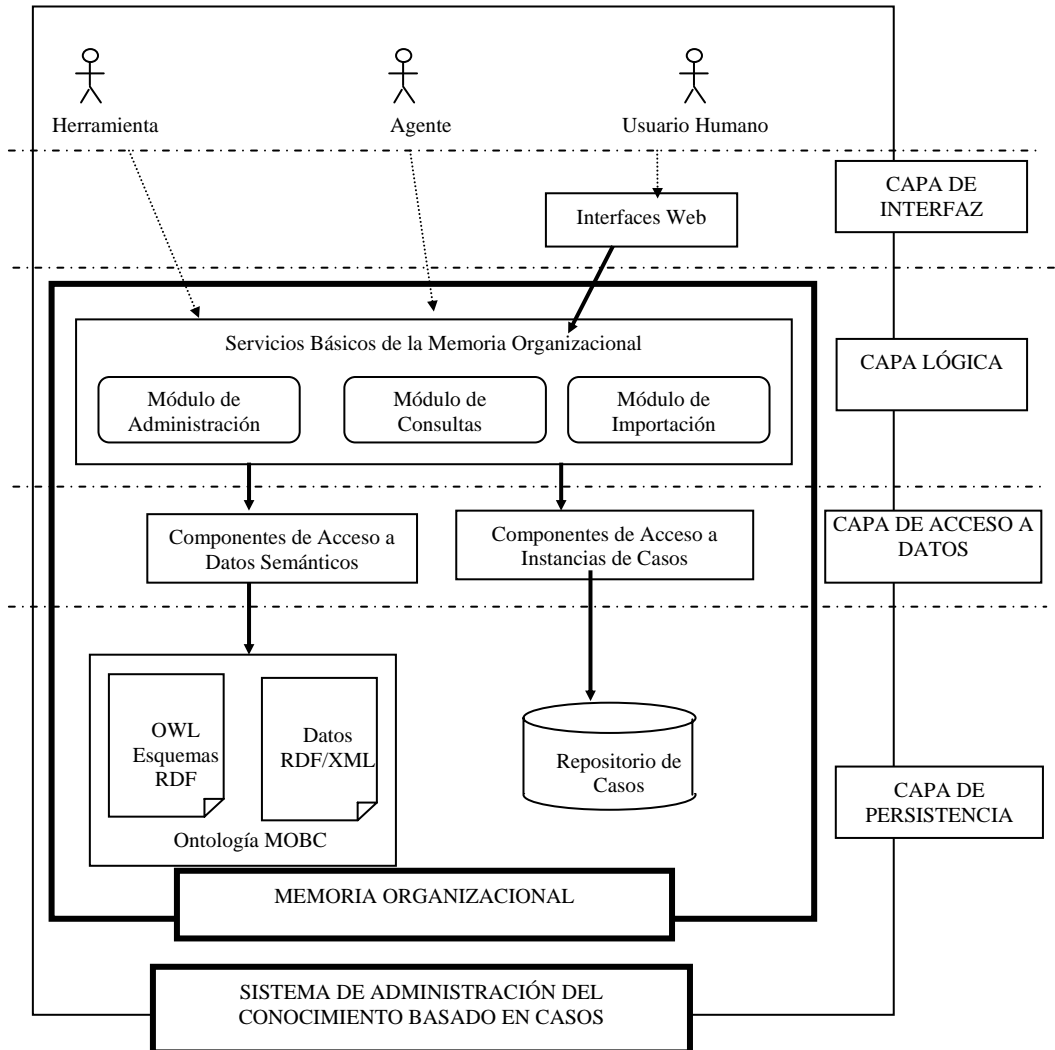


Figura 3. *Arquitectura de Memoria Organizacional Basada en Casos*

6. Un Caso Práctico: Sistema de Recomendación como Apoyo al Área de Aseguramiento de Calidad.

En esta sección mostraremos la aplicación del marco de MOBC en el diseño de un sistema de recomendación como apoyo al proceso de solución de problemas para el área de aseguramiento de calidad.

6.1 Introducción

Uno de los objetivos del proceso de apoyo de solución de problemas es la mejora continua del proceso de desarrollo. Cuando una organización decide implementar un sistema de gestión de calidad, es señal de que está dispuesta a abordar un proceso de autocrítica y aprendizaje y someter los procesos, métodos y sistemas a ciclos de mejora continua.

La mejora continua ha de ser un objetivo permanente en las organizaciones, dentro del convencimiento de que la no mejora significa retroceso. Este enfoque está claramente reflejado en la Norma ISO 9001 “Sistema de gestión de calidad. Requisitos”. El mantenimiento y la mejora continua de la capacidad del proceso deben favorecer la sistematización del conocimiento y posibilitar el aprendizaje en toda la organización. Uno de los requerimientos para la mejora continua es el del proceso de apoyo a la solución de problemas.

La norma ISO/IEC 12207 en el apartado relacionado al “Proceso de Aseguramiento de la Calidad”, más específicamente en lo relacionado a implementación del proceso (sección 6.3.1.4) establece: Cuando se detecten problemas o no conformidades con los requisitos del contrato, deberán documentarse y actuar como entrada al Proceso de Solución de Problemas (6.8). Se deberán preparar y mantener registros...de los problemas y de las soluciones. Para facilitar tal tarea de apoyo es importante contar con herramientas que:

- Mantengan una base de conocimiento donde se registren los problemas detectados, las soluciones tomadas y los resultados obtenidos es decir casos, de una forma que puedan ser procesados en forma automática (por ejemplo con tecnologías de Web Semántica).
- Recomiende posibles soluciones, basado en la similitud de casos y teniendo en cuenta el contexto [Dey01] del problema [Aam94] (Razonamiento basado en casos).

En la siguiente sección mostramos el diseño de un sistema de recomendación que da solución a este problema.

6.2 Sistema de Recomendación

Para diseñar el sistema de recomendación basado en la MOBC, debemos personalizar el modelo conceptual, definiendo la ontología de dominio de acuerdo al tipo de conocimiento que queremos almacenar en la memoria organizacional.

En el presente ejemplo el sistema deberá administrar conocimiento relacionado a “no conformidades”, según la especificación de la norma 9001, en el apartado 8.3. Dicha ontología nos facilitara la representación del conocimiento sobre problemas de no conformidad en forma de casos, su almacenamiento, disseminación y posterior reuso en la recomendación de la solución a nuevos problemas, basado en experiencias previas similares.

Una no conformidad se puede caracterizar por un tipo de defecto y su valor. La figura 4 presenta un modelo conceptual UML de la ontología para representar No conformidades. A la hora de codificar los tipos de defectos, podemos recurrir a trabajos de investigación que ya han realizado una clasificación de defectos de productos y servicios de software. Por ejemplo, Tervonen et al en su trabajo: “Quality knowledge capturing and reuse in software inspection”, presenta una clasificación de defectos en 13 categorías y sus respectivas subcategorías.

Usaremos esa clasificación, como base para codificar los tipos de defectos, pero se sugiere que sea adaptada a cada compañía de software, de acuerdo a sus necesidades. Cada “no conformidad” tiene asociada una acción correctiva, que se puede representar a través de una lista de tareas simples y específicas. Esta lista de tareas podrá incluir tanto acciones correctivas de la no

conformidad, como acciones correctivas del proceso de desarrollo, atacando las causas de los defectos, y favoreciendo el aprendizaje y la mejora continua.

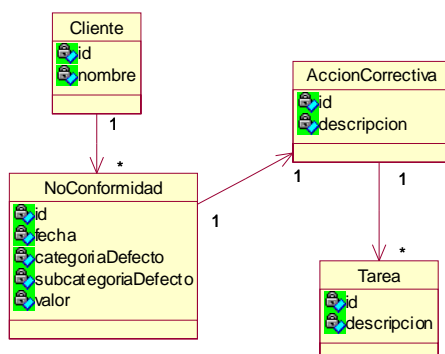


Figura 4. Modelo conceptual del dominio de No Conformidades

Asociando esta ontología de dominio al marco MOBC presentado en la sección 4, podemos representar las “lecciones aprendidas” o casos de no conformidades, con la estructura problema-solución, definida en la misma sección 4. En la figura 5 se muestra un ejemplo de caso que representa la no conformidad: “Falta soporte para la información de los e-mail de los clientes”, y su respectiva solución, o acción correctiva. El almacenamiento y tratamiento de las no conformidades usando una memoria organizacional basada en casos, permite su procesamiento automático, y potencia la difusión y reuso del conocimiento o lección aprendida que encierra.

<p>CASO: No conformidad debido a la falta soporte para la información de los e-mails de los clientes.</p>
<p>PROBLEMA: NoConformidad.id=21584, Noconformidad.fecha=22/02/2006 NoConformidad.categoriaDefecto = “data defect”, NoConformidad.subcategoriaDefecto = “data insufficient”, NoConformidad.valor = “e-mail cliente”.</p>
<p>SOLUCIÓN:</p> <ul style="list-style-type: none"> – Agregar ítem en el modelo de datos – Ampliar esquema en la base de datos – Agregar campo en la interfaz de usuario – Compilar e instalar el nuevo producto – Comunicar al cliente

Figura 5. Ejemplo de un caso de No conformidad almacenado en la MOBC

Además, el KMS deberá soportar funcionalidades de apoyo a todas las actividades que comprende un proceso de administración del conocimiento que son:

- **Captura y creación:** La captura y creación del conocimiento puede realizarse de dos formas: en forma manual (mediante el ingreso de las no conformidades informadas por el cliente), o en forma automática a través de herramientas de inspección integradas al KMS. La integración de herramientas al KMS es facilitado por las funcionalidades provistas por la arquitectura MOBC a través de sus módulos implementados como API's.
- **Búsqueda y acceso:** La utilidad de la memoria organizacional basada en casos reside en que el conocimiento almacenado puede ser reusado en la solución de problemas de no

conformidades similares. Para fortalecer este reuso, el marco provee un motor de razonamiento basado en casos, que recupera soluciones, teniendo en cuenta casos similares, y el contexto donde se aplica. La interpretación contextual es posible, porque a cada caso se le asocia los parámetros de contexto en el cual la solución es aplicable. Como se mostró en la sección 4 y 4.2.

- **Recomendación:** Usando el razonamiento basado en casos provisto por el marco, el sistema recomendará acciones correctivas a tomar, en forma pro-activa, interactuando con herramientas de inspección o desarrollo.

7. Conclusiones

En este trabajo se definió el marco conceptual y diseño de una base de conocimientos o memoria organizacional basada en casos. Para mostrar su flexibilidad y funcionalidad, se ejemplificó con un caso práctico: un sistema de recomendación como apoyo a la solución de problemas de no conformidades especificado en la norma ISO 9001.

La administración sistemática y automatizada del conocimiento organizacional en las empresas dedicadas al desarrollo de software es una ayuda importante para lograr la mejora continua de los procesos, propiciada por dicha norma. El modelo de memoria organizacional presentado en este informe es la base fundamental para la implementación de herramientas y sistemas de recomendación que gestionen dicho conocimiento en forma semántica, interoperativa y automatizada. Este enfoque estructurado y basado en tecnologías de Web Semántica y Razonamiento Basado en Casos, permitirá la recuperación de experiencias pasadas y lecciones aprendidas representadas a través de casos similares, y la recomendación de soluciones de buena calidad que tenga en cuenta el contexto del problema, sirviendo, de esta forma, como apoyo a la toma de decisiones dentro de la organización, y al aprendizaje y mejora continuas.

Varias son las líneas de avance futuras, entre ellas: la especificación ontológica, con mayor detalle, de la memoria organizacional en función del contexto, y la implementación del sistema diseñado con tecnologías de Web Semántica, tal como se hizo con el sistema de medición y evaluación [Ols05].

Agradecimientos

Este proyecto está parcialmente soportado por el Programa de Incentivos, en el proyecto 09/F037, así como por los proyectos PICT 11-13623 y PAV 127-5.

Referencias

- [Aam94] Aamodt, A. Plaza E. (1994) “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Comm. IOS Press, Vol. 7:1, pp.39-59.
- [Che03] Chen, H. & Wu, Z. (2003) On Case-Based Knowledge Sharing in Semantic Web. In: 15th IEEE Int. Conference on Tools with Artificial Intelligence (ICTAI'03)
- [Con96] Conklin, J.; (1996) “Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy. Group Decision Support Systems, <<http://www.gdss.com/ \DOM.htm>>
- [Dey01] Dey, A. (2001) “Understanding and Using Context”. Personal and Ubiquitous Computing 5(1).
- [Dog93] Dogson, M.; “Organizational Learning: A Review of Some Literatures”, <http://www.bmgt.umd.edu/Business/AcademicDepts/IS/learning/orglrn.html>.

- [Fen03] Fensel, D. (2003) “Ontologies: a silver bullet for Knowledge Management and Electronic Commerce”, 2nd edition; Springer-Verlag, Berlin, Germany .
- [Kol93] Kolodner J. (1993), “Case-based Reasoning”, Edit. Morgan Kaufmann
- [Mal96] Malhotra, Y (1996) Organizational Learning & Learning Organizations: An Overview
- [Mar03] Martín M., Olsina L. (2003) “Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System”, *In proceed. of IEEE Computer Society (1st Latin American Web Congress)* Chile, pp103-113, ISBN 0-7695-2058-8.
- [Mol04] Molina H., Papa F., Martín M. de los A., Olsina L. (2004) “Semantic Capabilities for the Metrics and Indicators Cataloging Web System”. In: *Engineering Advanced Web Applications*, Matera M. and Comai S. (Eds.), Rinton Press Inc., US, pp. 97-109, ISBN 1-58949-046-0.
- [Ols04] Olsina L., Martín M. (2004) “Ontology for Software Metrics and Indicators”, *Journal of Web Engineering*, Rinton Press, US, Vol 2 N° 4, pp. 262-281, ISSN 1540-9589.
- [Ols05] Olsina L., Papa F., Molina H. (2005) “Organization-Oriented Measurement and Evaluation Framework for Software and Web Engineering Projects”, In: *Lecture Notes in Computer Science of Springer, LNCS 3579*, Int'l Congress on Web Engineering, (ICWE05), Sydney, Australia, pp. 42-52.