Soporte a la Recomendación en Aseguramiento de Calidad usando Memoria Organizacional

Belén Rivera, María de los A. Martín y Luis Olsina

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

Resumen. Las lecciones aprendidas por las personas son una clase de conocimiento valioso para una organización (por ej., de software), a la hora de resolver situaciones similares (casos). Una forma de aprovechar mejor dicho conocimiento, de manera que dé soporte a la recomendación en la toma de decisiones, es contar con una memoria organizacional que lo administre y procese en forma consistente y eficaz. En este trabajo a partir de la ontología de memoria organizacional basada en casos, mostramos su utilidad en una herramienta de recomendación en aseguramiento de calidad para proyectos de medición y evaluación.

1 Introducción

Los sistemas de administración del conocimiento (KMS) tienen como objetivo administrar el conocimiento organizacional, de modo que luego pueda ser utilizado para aprender, resolver problemas y como apoyo en la toma de decisiones [3, 5]. Además promueven una mejora en tiempo y costo al permitir conocer los errores y aciertos del pasado para solucionar problemas del presente sin repetir los mismos errores, tomando ventaja de las lecciones aprendidas. Las empresas de software son organizaciones en donde las personas continuamente expanden sus capacidades para crear soluciones a problemas nuevos o recurrentes, y donde continuamente aprenden de sus acciones y experiencias.

En Ingeniería de Software/Web el aseguramiento de la calidad y en particular la medición y evaluación, son procesos de soporte claves para el desarrollo y mantenimiento de aplicaciones. Por lo tanto, la administración en forma consistente y eficaz del conocimiento relacionado a proyectos de evaluación es clave para la toma de decisiones y facilita la recomendación en nuevos proyectos, tomando ventaja de las experiencias anteriores y reusando las buenas prácticas. La mayoría de los KMS que existen en la actualidad capturan y almacenan el conocimiento en repositorios de documentos como manuales, informes, 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 el conocimiento, ocasiona pérdidas de tiempo y alta inversión en recursos humanos, ya que no contemplan mecanismos potentes de procesamiento semántico y automático de dicho conocimiento. Una de las formas de resolver este problema es almacenar el conocimiento en forma estructurada, por ejemplo, en lo que se denomina memoria organizacional basada en casos (MOBC).

A partir de la ontología de MOBC [9] la contribución del presente trabajo consiste

en ilustrar su utilidad para el desarrollo de una herramienta de recomendación en aseguramiento de calidad; es decir, está basada en ontologías y servicios Web para asegurar interoperabilidad, consistencia, generalidad y potencia de procesamiento.

El resto del artículo se organiza de la siguiente manera. En la sección 2 se resume la ontología de memoria organizacional basada en casos. En la sección 3 se muestra el sistema de recomendación en aseguramiento de calidad usando la memoria organizacional basada en casos. Para ilustrar la utilidad de la herramienta, en la sección 4 se describe con detalle el proceso de recomendación en la etapa de especificación de requerimientos para proyectos de medición y evaluación. Finalmente se delinean las conclusiones y oportunidades de mejora.

2. Ontología de Memoria Organizacional basada en Casos

La ontología de memoria organizacional es una ontología de nivel genérico (para mayor detalle ver [9]), a partir de la cual se pueden formular otras representaciones para ser aplicadas a dominios específicos. Por un lado se define la ontología genérica de MOBC y, por otro, para caracterizar los casos de acuerdo al conocimiento de un dominio específico, se deberán definir las ontologías de dominio y de contexto [10].

En esta sección describimos los principales conceptos de la ontología de MOBC, que se ilustran en la Fig. 1. Una memoria organizacional (*OrganisationalMemory*) es un repositorio que almacena la totalidad del conocimiento formal e informal presente en una organización y por lo tanto tiene una o más bases de conocimiento *(KnowledgeBase)*. Un tipo de base de conocimiento es la base de conocimiento basada en casos *(CaseKnowledgeBase)* que almacena el conocimiento adquirido en experiencias pasadas, buenas prácticas, lecciones aprendidas, heurísticas, etc. de distintos dominio; es decir, almacena casos. Un caso es una pieza *contextualizada* de conocimiento que representa una experiencia y se compone de un problema *(problem)* y su solución *(solution)*. Teniendo en cuenta esta definición, es fundamental en toda memoria organizacional guardar la información del contexto *(context)* donde ocurre cada caso. Una forma de modelar contexto ha sido desarrollada en [10].

La representación del conocimiento a través de casos, facilita el reuso del conocimiento adquirido en situaciones de problemas similares pasados para ser aplicado a un nuevo problema [1]. Tradicionalmente, existen varios métodos para representar casos, que van desde los no estructurados hasta representaciones completamente estructuradas y automáticamente procesables [2]. Nosotros optamos por la última estrategia, donde se especifican formalmente el significado de los conceptos involucrados y, por lo tanto, son automáticamente procesables.

Los problemas y las soluciones se describen a través de variables de características (*Feature*): ya del problema (*ProblemFeature*), ya de la solución (*Solution Feature)*. Debido a que los casos almacenados en la memoria se refieren a conocimiento de un dominio específico, estas características están formalmente definidas por un término en la ontología de dominio relacionada a la memoria organizacional (en nuestro caso, usaremos la ontología de métricas e indicadores especificada en [8]). El proceso de razonamiento basado en casos consiste en asignar valores a las variables de características del problema y encontrar los valores adecuados para las instancias de la solución, a través de criterios de evaluación de similitud de casos.

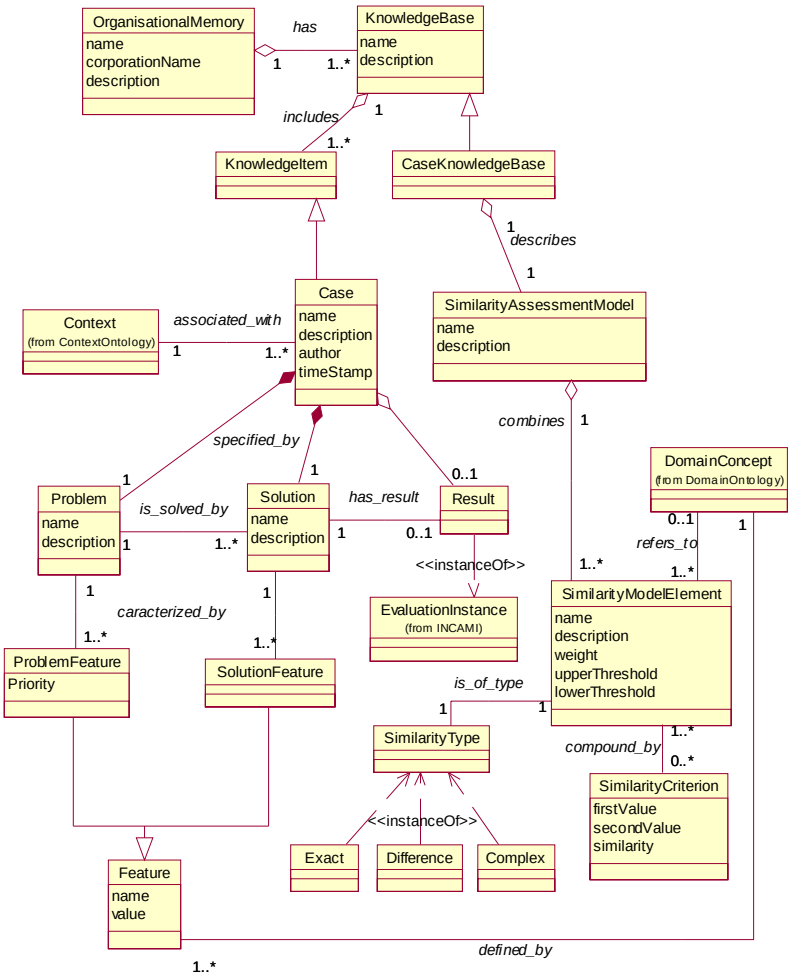

Figura 1. *Modelo conceptual de la ontología de memoria organizacional basada en casos*

La mayoría de las aplicaciones de CBR se han enfocado en dominios de problemas específicos [7]. Sin embargo, para que un sistema CBR sea útil a una organización, debería ajustarse a las principales fuentes de conocimiento que puede ser de distintos dominios y, por lo tanto, necesitan funciones de similitud apropiadas a cada base de casos. Así, para cada base de conocimiento basado en casos, se debe especificar un modelo de evaluación de similitud *(SimilarityAssessmentModel)*, la que combina varios elementos de similitud *(SimilarityModelElement)*, uno para evaluar cada característica del caso. Tradicionalmente, la similitud entre un caso recuperado R y uno nuevo C, se define como la suma de las similitudes entre los valores de sus características constituyentes multiplicados por sus pesos de relevancia relativa:

$$Similitud\ (R, C) = \sum_{f \in F} w_f \times sim_f\ (f_R , f_C) \qquad (1)$$

donde w_f es el peso de relevancia de la característica *f*, y sim_f es la función de

medida de similitud de una característica especifica *f*.

Por lo tanto, para proveer una representación adecuada de la similitud, es necesario representar tanto los pesos de relevancia como la descripción de la función de similitud para una característica específica. Los pesos se representan como un atributo dentro de cada elemento de similitud, y la función de similitud se restringe a tres tipos generales de funciones de similitud: *Exact, Difference y Complex*. La función de similitud *Exact*, devuelve 1 si los valores de característica son iguales, y 0 en otro caso. La función de similitud *Difference*, es inversamente proporcional a la diferencia entre los valores de las características. La función de similitud *Complex*, resuelve la similitud para todas aquellas situaciones donde las dos funciones de similitud anteriores no son aplicables. Por ejemplo, la diferencia semántica entre dos términos sinónimos, que no necesariamente son totalmente iguales ni totalmente distintos. Si el número de valores posibles de la característica es finito, es posible guardar la medida de similitud para cada combinación de valores posibles. En nuestro modelo, estos parámetros están representados en la clase *(SimilarityCriterion)*. La elección de estos tipos de funciones se basa en el análisis de trabajos de investigación en el área [1, 4].

3. Sistema de Recomendación para la Herramienta de Medición y Evaluación en Aseguramiento de Calidad C-INCAMI_Tool.

Una aplicación de la MOBC es el sistema de recomendación diseñado para la herramienta C-INCAMI_Tool [11]. C-INCAMI es un marco que da soporte a todos los procesos involucrados en la medición y evaluación de requerimientos no funcionales en proyectos de software y Web. Está basado en una ontología de métricas e indicadores [8] que define de forma explícita los conceptos, atributos y relaciones principales que se utilizan en las actividades de medición y evaluación.

Una organización que busque medir y evaluar un proyecto de software de un modo eficaz, debe hacerlo orientado a un propósito, por lo cual se deben especificar requerimientos no funcionales a partir de la identificación de una necesidad de información y el contexto de información relevante al proyecto. Luego, se deben diseñar y seleccionar un conjunto específico de métricas útiles para el propósito definido. Por último, se deben interpretar los valores obtenidos por medio de indicadores, para evaluar el grado de satisfacción que proporcionan los requerimientos, y finalmente, sacar conclusiones y dar recomendaciones apropiadas. Este marco y su herramienta está formada por cinco componentes principales que son: (*1) Definición y Especificación de Requerimientos no Funcionales; (2) Especificación del Contexto del Proyecto; (3) Diseño y Ejecución de la Medición; (4) Diseño y Ejecución de la Evaluación; (5) Análisis y Recomendación.*

3.1. Arquitectura del Sistema de Recomendación

El sistema de recomendación propuesto robustece a INCAMI_Tool en las primeras etapas de un proyecto de medición y evaluación, donde se requiere de la experticia en el diseño y donde la disponibilidad de recomendaciones desde una base de conocimientos puede ser de utilidad.

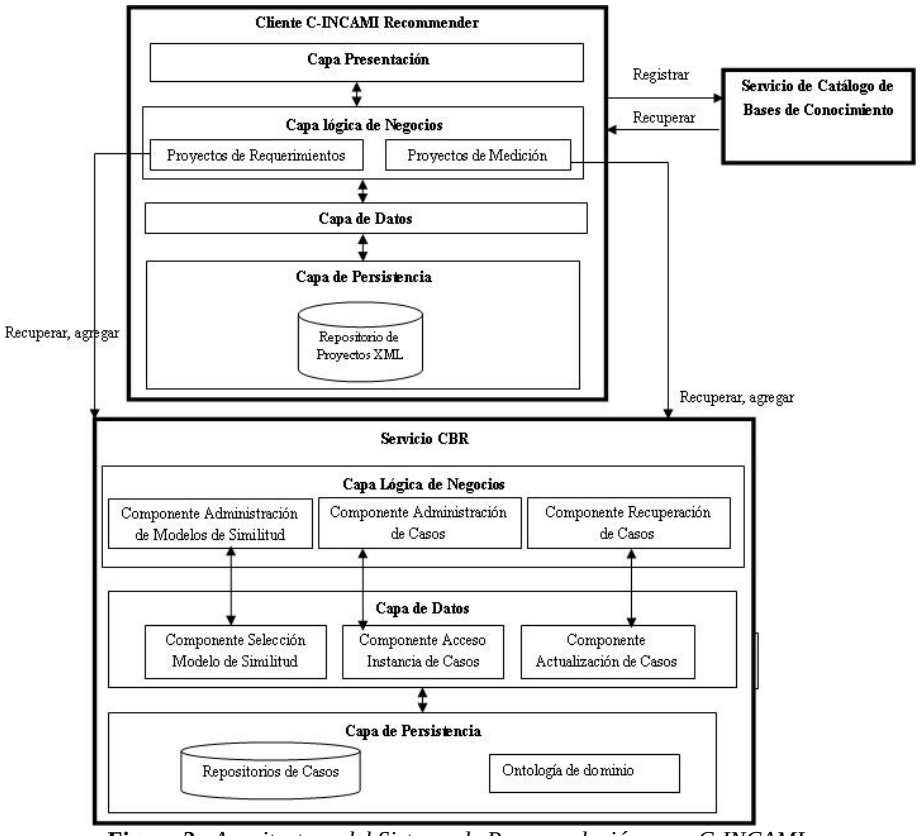

Figura 2. *Arquitectura del Sistema de Recomendación para C-INCAMI*

En la etapa de especificación de requerimientos, el sistema recomendará un árbol de requerimientos apropiado para las características del proyecto, basado en la similitud de casos anteriores guardados en la memoria organizacional –en la sección 4 se muestra con detalle el mecanismo de CBR usado para soportar tal recomendación. En la etapa de diseño de la medición, el sistema recomendará las métricas apropiadas para la medición de cada atributo, también usando casos anteriores guardados.

El sistema de recomendación para C-INCAMI, sigue el modelo arquitectónico distribuido en capas, basado en servicios Web y desarrollado con tecnologías de Web semántica. Se estructura en dos grandes componentes (Fig. 2): Primero, *el módulo de recomendación* en sí, que es el cliente del sistema, denominado *C-INCAMI Recommender*, el cual permite: (i) la definición y especificación de proyectos de requerimientos; (ii) la especificación del contexto del proyecto; y (iii) el diseño de la medición. Durante estas etapas, el sistema en forma proactiva recomienda al usuario posibles árboles de requerimientos y posibles métricas, de acuerdo a las características del proyecto y su contexto. Para cumplir estas recomendaciones el módulo usa los servicios de la memoria organizacional basada en casos. Segundo, *el módulo de la memoria organizacional*, que se encuentra distribuida en varias bases de conocimiento basadas en casos, cada una implementada como un servicio Web

llamado servicio CBR. En este caso, se usan sólo dos bases de conocimientos, una base de conocimientos de árboles de requerimientos y otra base de conocimientos de métricas. A su vez, otro servicio Web, el denominado *Servicio de Catálogo de Bases de Conocimiento* mantiene un catálogo de los nombres y las URLs de todas las bases de conocimiento distribuidas basadas en casos.

3.1.1 El Módulo de Recomendación

El estilo arquitectónico para el módulo de recomendación, es de cuatro capas:

- Capa de Presentación: interfaz de usuario que permite al evaluador crear proyectos de requerimientos o de medición.
- Capa lógica de negocios: recibe las peticiones de la capa de presentación. Se conforma de: *El componente de administración de proyectos de Requerimientos,* que crea proyectos de requerimientos para un proyecto de medición y evaluación en particular. Se comunica con el módulo de memoria organizacional a fin recomendar un árbol de requerimiento para el proyecto actual. Y *el componente de administración de proyectos de medición,* que crea proyectos de medición para un proyecto de requerimientos seleccionado. Se comunica con el módulo de memoria organizacional a fin recomendar métricas para los atributos en cuestión.
- Capa de datos: compuesta por los componentes que sirven para acceder a los datos de la aplicación.
- Capa de persistencia: compuesta por todos los paquetes y clases relacionadas de manera de almacenar datos y los sistemas gestores de bases de datos.

3.1.2 El Módulo de Memoria Organizacional

El *Servicio CBR* (ver parte inferior de la Fig. 2) implementa los servicios Web básicos de la MOBC y puede ser replicado; esto permite que la MOBC pueda ser distribuida en varias bases de conocimiento y accedida por cualquier cliente que gestione los tipos de conocimiento que ésta administra. El *Servicio CBR* es el encargado de administrar la base de casos y resolver consultas de casos similares a través de sus dos operaciones definidas que son *agregar* y *recuperar*.

Para implementar estas operaciones el servicio tiene acceso a la ontología de métricas e indicadores, que constituye la ontología de dominio asociada a esta base de conocimiento. Además, para resolver cuál es el mejor caso, cada base de conocimiento tiene asociado un modelo de similitud, que personaliza la lógica del razonamiento basado en casos. Este módulo no posee capa de presentación debido a que sólo presta servicios a aplicaciones clientes. Las capas que lo componen son:

- Capa lógica de negocios: presenta la misma funcionalidad que la descripta para el cliente. Se compone de los siguientes componentes:
  - *Componente de Administración de Modelos de Similitud:* el modelo de similitud puede ser distinto para cada base de conocimiento. Esto permite adecuar el motor de razonamiento al tipo de experiencias que almacena la base de casos a través de la selección de un modelo de similitud.

- *Componente de Recuperación de Casos*: a partir de un nuevo caso proveniente del cliente y, considerando el tipo de base de conocimiento para ese caso, realiza la evaluación de similitud del nuevo caso contra aquellos almacenados en la memoria. Devuelve la solución del caso más similar, como ejemplificaremos en la Sección 4.
- *Componente de Administración de Casos:* permite la actualización y agregado de nuevos casos a la memoria.

- Capa de datos: compuesta por los componentes que sirven para acceder a los datos de la aplicación.
- Capa de persistencia: almacena de manera persistente la información de casos además del conocimiento del dominio.

Si observamos la Fig. 2, el cliente *C-INCAMI Recommender* podría ser cualquier aplicación, agente u otro servicio Web que necesite resolver un problema similar a los almacenados en la base de casos. La flexibilidad de la arquitectura permite que pueda ser usada por cualquier sistema de recomendación, instanciando adecuadamente la ontología de dominio y el modelo de similitud de acuerdo al ámbito de la aplicación. Para ilustrar la utilidad de esta arquitectura se muestra un caso práctico.

4. Recomendación en la Etapa de Especificación de Requerimientos

El sistema de recomendación se basa en la MOBC que administra casos relacionado a proyectos de medición y evaluación, que están caracterizados con términos de la ontología métricas e indicadores [8]. Esta ontología incluye la definición de *ProjectPurpose, CalculableConcept, ConceptModelType, Userview, EntityCategory, Entity* entre otros conceptos.

Para adecuar el motor CBR a nuestra aplicación, se debió definir también el modelo de similitud. En la tabla 1 se muestra el modelo de similitud usado a partir de la ontología (ver Fig. 1). Para cada *Feature* que caracteriza a un caso debemos establecer su peso de preponderancia relativa y el tipo de función de similitud. Estas decisiones de diseño las debe tomar un experto, teniendo en cuenta qué características se consideran más relevantes desde el punto de vista de la similitud, para evaluar en definitiva la similitud global de dos casos.

Tabla 1. *Ejemplo de modelo de similitud para una la base de casos*

Description	Feature	SimilarityType	Weight
Propósito de la evaluación	Purpose	Complex	0.1
Concepto calculable a evaluar, por ej. calidad, etc.	CalculableConcept	Exact	0.3
Tipo de modelo, por ej. ISO o Propio	ConceptModelType	Exact	0.05
Punto de vista del usuario	Userview	Exact	0.15
Tipo de categoría de entidad a evaluar	EntityCategory	Exact	0.25
Tipo de entidad a evaluar	Entity	Exact	0.15

Como podemos observar en la tabla 1, las características *CalculableConcept, ConceptModelType, Userview, EntityCategory* y *Entity* tienen funciones de similitud tipo *Exact* cuyo cálculo es trivial; en cambio, para la característica *Purpose*, el cálculo de similitud no es trivial debiendo proporcionarse las medidas de similitud para cada par de valores posibles.

Tabla 2. *Ejemplo de definición de similitud compleja*

	Evaluate	***Estimate***	***Improve***
Evaluate	1	0.8	0.4
Estimate	0.8	1	0.3
Improve	0.4	0.3	1

Suponiendo que los valores posibles para la característica *Purpose* son: *Evaluate*, *Estimate* e *Improve*, y que un experto determina que hay mayor similitud entre los valores *Evaluate y Estimate* (valorada subjetivamente en 0.8) que entre los valores *Estimate* e *Improve* (0.3), y que entre los valores *Evaluate* e *Improve* (0.4), la tabla 2 muestra esta definición de similitud compleja, donde se puede ve la mayor similitud semántica entre los valores *Evaluate* y *Estrimate*. Con estas especificaciones, los casos se representan con los valores de todas las *features* que lo caracterizan, y su respectiva solución como se ilustra en el caso de ejemplo que se muestran en la Fig. 3. Dicho caso representa al problema de obtener un árbol de requerimientos para una necesidad de información específica y la solución llevada a cabo.

CASO 1: Obtención de Árbol de requerimientos para evaluar la confiabilidad de enlaces para páginas estáticas del sitio Web www.unlpam.edu.ar

PROBLEMA:

Purpose	Evaluate
CalculableConcept	Link Reliability
ConceptModelType	ISO
UserView	Evaluator
EntityCategory	Software Product
Entity	www.unlpam.edu.ar

SOLUCIÓN:

1. *Link Reliability*
   - *1.1. Internal Broken Links*
   - *1.2 External Broken Links*
   - *1.3 Invalid Links*

Figura 3. *Ejemplo de representación de un caso de diseño de árbol de requerimientos.*

Para ilustrar mejor este modelo de similitud, en la tabla 3 se muestran dos casos típicos tomados de la base de casos de proyectos de medición y evaluación, con los valores de sus características, el tipo de función de similitud y el valor de similitud para cada característica calculado. Para calcular la similitud global de los dos casos, se aplica la fórmula (1), y reemplazando los pesos definidos en el modelo de similitud (tabla 1) y los valores de similitud local de cada característica (tabla 3), resulta:

Similitud (C1, C2) = 0.1×0.8 + 0.3×1 + 0.05×0 + 0.15×1 + 0.25×1+0.15 × 0 = 0.78

Tabla 3. *Ejemplo de similitud entre las características de dos casos.*

Feature	**similarityType**	**Caso 1**	**Caso 2**	**Similitud**
Purpose	Complex	Evaluate	Estimate	0.8
CalculableConcept	Exact	Link Reliability	Link Reliability	1
ConceptModelType	Exact	ISO	Propio	0
Userview	Exact	Evaluator	Evaluator	1
EntityCategory	Exact	Software Product	Software Product	1
Entity	Exact	www.unlpam.edu.ar	www.itc.edu.ar	0

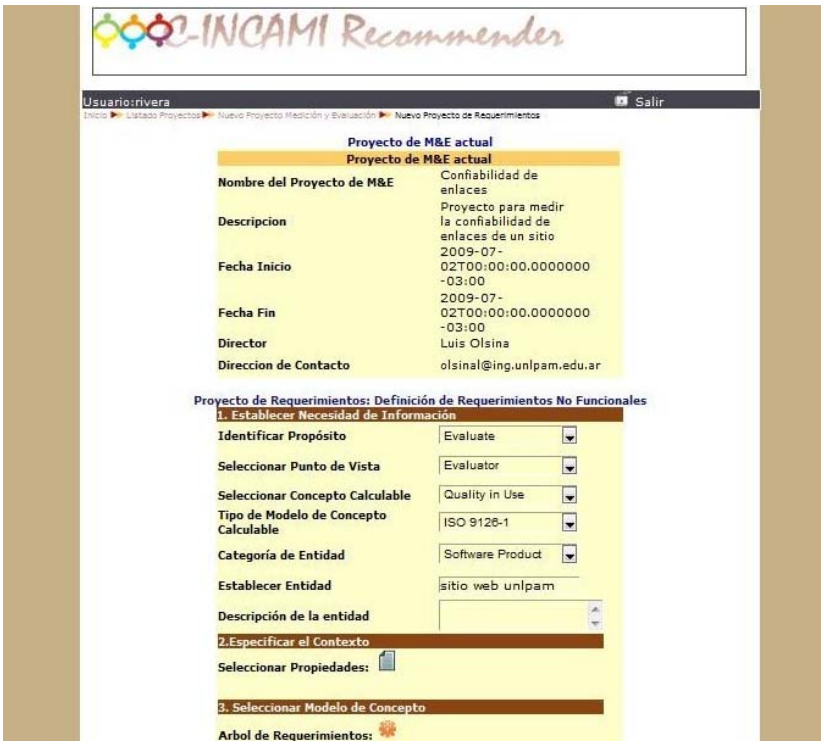

Figura 4. *Una pantalla de la herramienta C-INCAMI Recommender.*

Un nuevo proyecto de medición y evaluación puede tomar ventajas de la memoria organizacional basada en casos, recuperando la información de diseño del proyecto más parecido y luego adaptándolo al nuevo caso.

Cuando el usuario evaluador de la herramienta desarrollada (ver una pantalla en Fig. 4) crea un proyecto de medición y evaluación, debe especificar los requerimientos no funcionales, es decir, identificar el propósito de la medición (*ProjectPurpose*), el punto de vista (*Viewpoint*), el concepto calculable (*CalculableConcept*), categoría de entidad (*EntityCategory*), la entidad propia a evaluar (*Entity*) y las propiedades que definen el contexto del proyecto. Al momento de ingresar el árbol de requerimientos para la medición, en lugar de diseñarlo completamente podría emplear al sistema de recomendación, que basándose en la similitud con los casos anteriores, le recomendará una solución adecuada, usando el motor de razonamiento basado en casos de la memoria organizacional, solicitada al módulo *Servicio CBR*.

De manera similar, usando la base de conocimiento de métricas, con su propio modelo de similitud, el sistema de recomendación ayuda a seleccionar métricas adecuadas en la etapa de diseño de la medición de C-INCAMI.

6. Conclusiones

Administrar el conocimiento organizacional representa una ventaja clave en toda

empresa de software. Para que dicha administración sea eficiente es importante contar con una memoria organizacional que de soporte a la captura, estructuración, reuso y diseminación del conocimiento creado por sus empleados. La utilidad de una memoria organizacional radica en que el conocimiento que almacena pueda ser fácilmente compartido entre sistemas dentro de la organización o entre organizaciones colaborativas.

Particularmente en Ingeniería de Software y Web, el aseguramiento de la calidad y como consecuencia la medición y evaluación, son procesos de soporte claves para el desarrollo y mantenimiento de aplicaciones de software. Por lo tanto la administración en forma consistente y eficiente del conocimiento relacionado a proyectos de medición y evaluación es muy importante para la toma de decisiones y facilita la recomendación en nuevos proyectos, tomando ventaja de las experiencias y lecciones aprendidas.

En este trabajo mostramos cómo se puede implementar una MOBC que administre en forma consistente y eficiente el conocimiento relacionado a proyectos de medición y evaluación y, además, presentamos una arquitectura flexible de memoria organizacional basada en casos, que está centrada en ontologías y servicios Web. Estas propiedades hacen que la MOBC se pueda integrar fácilmente a los sistemas de información de una organización, a través de módulos de inspección y/o recomendación. Para ilustrar esto, describimos una herramienta de recomendación para C-INCAMI. Como trabajo futuro, diseñaremos un experimento a fin de evaluar la ganancia en la eficacia de una estrategia centrada en administración del conocimiento con memoria organizacional, con respecto a una tradicional. Para tal efecto, se usará la herramienta con y sin el módulo de recomendación.

Referencias

1. 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.
2. 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)
3. Conklin, J. (1996) "Designing Organizational Memory: Preserving Intelectual Assets in a Knowledge Economy". <http:\\www.gdss.com \DOM.htm>
4. Coyle, L; Doyle, D; Cunningham, P.(2004) *Similarity for CBR*. TR TCDCS-2004-25, Dublin.
5. Dogson, M.; *Organizational Learning: A Review of Some Literatures*, http:\\www.bmgt.umd.edu\Business\ AcademicDepts\ IS\learning\orglrn.htlm.
6. Fensel, D. (2003) Ontologies: a silver bullet for Knowledge Management and Electronic Commerce, 2nd edition; Springer-Verlag, Berlin, Germany.
7. Kolodner J. (1993), "Case-based Reasoning", Ed. Morgan Kaufmann
8. Martín M., Olsina L. (2003) *Toward an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System*, IEEE CS Press, 1st LA-Web, Chile, pp103-113.
9. Martín M., Olsina L. (2009) Added Value of Ontologies for Modeling an Organizational Memory. *Chapter 10, In the Book "Building Organizational Memories: Will You Know What You Knew?" John Girard (Ed). IGI Global*, USA, ISBN: 978-1-59904-540-5.
10. Molina H., Olsina L. (2007) *Towards the Support of Contextual Information to a Measurement and Evaluation Framework*. In 6th Int'l Conference on the Quality Information and Communications Technology (QUATIC), IEEE CS Press, pp.154–163. Lisbon, Pt.
11. Olsina, L; Papa, F.; Molina, H. (2007). How to Measure and Evaluate Web Applications in a Consistent Way. In: *Springer, HCI Series, Rossi, Pastor, Schwabe & Olsina (Eds.) Web Engineering: Modelling and Implementing Web Applications.* pp. 385–420.