

Un Marco de Referencia Metodológico para un Sistema Basado en Conocimiento Usando MDA

Pablo Valera^{1,2}, Dinarle Ortega², María Guevara²
pjvalera@yahoo.com, dinarleortega@gmail.com, mguevara@uc.edu.ve

¹ Departamento de Estudios Básicos, Universidad José Antonio Páez, Valencia, Venezuela

² Departamento de Computación, FACYT, Universidad de Carabobo, Valencia, Venezuela

Resumen: El presente trabajo muestra una revisión bibliográfica de los Sistemas Basados en Conocimiento (SBC), identificando la problemática de la ausencia de un lenguaje de representación estándar para su modelado, así como la trazabilidad clara entre sus modelos conceptuales y de diseño.

Posteriormente, se busca aplicar un enfoque de una Arquitectura Dirigida por Modelos (MDA, en inglés) a una metodología ampliamente conocida para el desarrollo de sistemas basados en conocimiento (COMMONKADS), y de esta manera generar un modelo independiente (*Platform Independent Model*, PIM) y un modelo específico de la plataforma (*Platform Specific Model*, PSM) que permitan la transformación así como la trazabilidad entre el modelo conceptual y de diseño de los SBC, para convertirlos de esta forma en los actores centrales del proceso de desarrollo. En particular, se genera un PSM para Java exhibiendo algunas transformaciones que seguidamente serán aplicadas a un caso de estudio. El PIM y el PSM obtenidos son diagramados en UML buscando unificar o proveer un lenguaje de representación alternativo para el modelado de este tipo de sistemas. Finalmente, se instancian los resultados en un caso de estudio para ser validados en el modelado de un sistema para el pronóstico de cáncer de seno.

Palabras Clave: MDA; SBC; COMMONKADS.

Abstract: This paper presents a literature review of Knowledge Based Systems (KBS), identifying the problem of the lack of a standard language representation for modeling and clear traceability between their conceptual and design models.

Subsequently, it seeks to apply an approach of Model Driven Architecture (MDA) to a widely known methodology for developing knowledge based systems (COMMONKADS), and thus generate a Platform Independent Model (PIM) and a Platform Specific Model (PSM) that allow the transformation and traceability between the conceptual and design models of the KBS, turning them this way in the central actors in the development process. In particular, a PSM is generated for Java exhibiting some transformations that will be applied to a case of study. PIM and PSM obtained are diagrammed in UML seeking to unify or provide an alternative representation language for modeling such systems. Finally, the results are instantiated in a case of study to be validated in the modeling of a breast cancer prediction system.

Keywords: MDA; KBS; COMMONKADS.

I. INTRODUCCIÓN

El quehacer económico y social está resultando cada vez más y más orientado al conocimiento, cobrando relevancia día a día el término, “sociedad de la información”. De acuerdo a Peter Druker [1] “El conocimiento es el único recurso significativo hoy en día”. Es un activo cada vez más apreciado por la sociedad contemporánea porque permite capturar la experiencia adquirida de las personas a través del tiempo, soportando de esta forma la toma de decisiones y facilitando el flujo de trabajo de cualquier empresa.

En particular, en el área de la informática se identifica al conocimiento como un elemento significativo para mejorar el

ejercicio de diversas labores, por medio de su representación y empleo para la sistematización de las tareas. Se procura crear una aplicación mecanizada que almacene, explote y actualice los conocimientos que los especialistas en un campo pueden aportar para la solución de problemas. Para lograr esto, se ha planteado la creación de “Sistemas Basados en Conocimiento” (SBC) correspondientes a un dominio específico, utilizando los aportes de la inteligencia artificial para propiciar el aprendizaje, razonamiento, representación y comunicación de éste conocimiento.

Los SBC, introducidos a principios de los años 70's como sistemas expertos desde el campo de la inteligencia artificial (IA), fueron originalmente diseñados para extraer

conocimiento de un humano experto en el dominio y codificar éste como reglas en una base de conocimiento (BC). No obstante, ha habido un cambio de paradigma en la Ingeniería del Conocimiento (IC), donde el enfoque de “extracción/transferencia” ha sido sustituido por el enfoque de “modelado”. Los modelos son usados para proveer una abstracción o simplificación acerca de la realidad y a través de los mismos, el enfoque del humano experto resolvidor del problema es modelado y usado en el desarrollo del SBC.

En la actualidad las metodologías de la IC están suficientemente maduras y ofrecen un conjunto de actividades y técnicas que abarcan el análisis y el posterior modelado conceptual del sistema a ser desarrollado de una manera adecuada [2]. Entre ellas se destacan COMMONKADS, MIKE, y PROTÉGÉ-II. De igual forma, se ha intentado también adaptar muchas técnicas de Ingeniería de Software (IS) a la IC, para mejorar aún más el desarrollo de los SBC, sin embargo, todavía queda camino por recorrer. En el caso de las metodologías mencionadas, éstas adolecen de una falta de rigor en cuanto a proveer elementos para la trazabilidad del modelado conceptual al diseño del sistema, tal y como lo propone la IS [3][4]. Adicionalmente, se aprecia que en la mayoría de los casos, varios lenguajes de modelado son aplicados en un proyecto. Por consiguiente, se experimenta la limitación de compartir y reusar el diseño de los SBC, por no haber un lenguaje de modelado ampliamente aceptado.

Esta limitación constituye una amenaza para los sistemas del futuro, debido a que serán diseñados para trabajar juntos con otras aplicaciones como parte de los sistemas de las empresas y requerirán el intercambio de reglas entre herramientas de modelado y motores de inferencia. La mayoría de estos sistemas son o serán construidos usando lenguajes de programación orientada a objeto para soportar técnicas de programación estandarizadas. Como no se conoce una forma estándar de modelar sistemas basados en conocimiento usando lenguajes de modelado de ingeniería del conocimiento, se plantea la necesidad de extender el uso de los lenguajes de modelado estandarizados de la ingeniería del software [2][3][5]. De este modo, sería posible integrar, reusar e interoperar entre los sistemas de las empresas y los motores de inferencia.

El planteamiento anterior evidencia la importancia de proponer estrategias de trabajo que potencien la reutilización a un alto nivel de abstracción. La Arquitectura Dirigida por Modelos (MDA en inglés) se ha convertido en una iniciativa para propiciar soluciones bajo este contexto. En este sentido, esta investigación propone la aplicación del enfoque MDA a una de las metodologías ampliamente conocida para el desarrollo de sistemas basados en conocimiento (COMMONKADS), y de esta manera generar un modelo independiente de la plataforma (*Platform Independent Model*, PIM) y un modelo específico de la plataforma (*Platform Specific Model*, PSM) que permitan la transformación así como la trazabilidad entre el modelo conceptual y de diseño de los SBC, para convertirlos de esta forma en los actores centrales del proceso de desarrollo, antes que el código.

El presente trabajo está estructurado de la siguiente manera: en la Sección II y III hablaremos brevemente acerca del estado del arte de los Sistemas Basados en Conocimiento y una de sus metodologías. Para posteriormente, en la Sección IV conversar

sobre MDA y luego aplicarlo en la Sección V, enfocándonos sobre una de las metodologías de SBC, donde se obtendrá un PIM y un PSM que serán validados subsiguientemente en la Sección VI. Finalmente, son presentadas nuestras conclusiones en la Sección VII.

II. LOS SISTEMAS BASADOS EN CONOCIMIENTO Y SU ENFOQUE

El *Object Management Group* (OMG) define un SBC como [6]: “Un Sistema Basado en Conocimiento, también conocido como sistema experto, es un software que tiene algún conocimiento o experticia acerca de un dominio reducido y es implementado tal que su base de conocimiento (BC) y arquitectura de control están separados. Los SBC frecuentemente incluyen procesamiento inferencial (opuesto al procesamiento algorítmico), explicación del razonamiento de los usuarios y la generación de resultados no únicos”. De la consideración de esta definición, podríamos decir que el conocimiento específico de un dominio es representado en una base de conocimiento, el cual es utilizado en el proceso de razonamiento del motor de inferencia para tomar decisiones relacionadas al dominio del problema. Esto es una característica única en un SBC, comúnmente utilizada para definirlos.

Las fases del proceso para construir un SBC en la ingeniería del conocimiento son en general: análisis de requerimientos identificando el alcance del SBC, diseño del sistema a través de la definición de las fuentes expertas en conocimiento y como representarlas, adquisición del conocimiento del experto a través de las diferentes técnicas de extracción y construcción para la base del conocimiento con las instancias del dominio, codificación, prueba, implementación y mantenimiento del sistema. Asimismo, la IS describe un conjunto de fases para el desarrollo de sistemas de software: Análisis de requisitos, Diseño del Sistema. Y, al comparar los procesos de desarrollo de la IS y la IC, se observa que la Ingeniería del Conocimiento presenta una fase adicional: “adquisición de conocimiento” Figura 1.

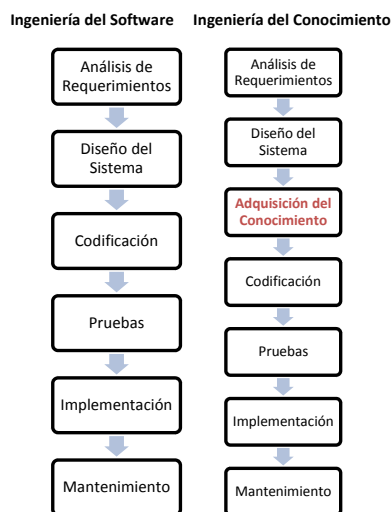


Figura 1: Comparación de los Procesos de Desarrollo para la IS y la IC [5]

La fase de adquisición de conocimiento es vital en el desarrollo del SBC debido a que el mismo es diseñado alrededor del conocimiento del experto en el dominio para la resolución de un problema en una tarea en particular, tal como un diagnóstico médico, una evaluación para un estudiante, etc. El conocimiento adquirido es entonces utilizado para poblar la base del conocimiento en forma de reglas con las cuales el sistema realizará el razonamiento.

La necesidad de administrar el conocimiento y las reglas del negocio a través de la tecnología ha causado que los SBC sean implementados en nuevos dominios y las capacidades de estos nuevos sistemas hayan sido explotadas para gestionar las competencias del ser humano. Algunos ejemplos son: diseño asistido para la arquitectura del software, gestión hospitalaria o clínica, manejo de riesgos en préstamos bancarios, asesoramiento en la bolsa de valores, entre otros. Más aún, son de gran valor ante la amenaza de rotación o cambio de personal y la pérdida del conocimiento (acumulación de hechos, eventos, procedimientos en la memoria del empleado que permiten la realización del trabajo y encontrarle sentido a los sucesos que pasan a su alrededor). Algunos autores [3][5][7], argumentan que esta “amenaza” es uno de los principales motores detrás de la aparición del enfoque en la gestión del conocimiento.

Por lo tanto, mientras los SBC tradicionales eran considerados aplicaciones estándares o tecnología de investigación en los laboratorios, hoy en día han resultado parte de los sistemas de información de las empresas, demandando aplicaciones comerciales ampliamente aceptadas por la industria. Gracias a esta alta aceptación, existe una fuerte demanda por parte de las empresas y los desarrolladores de estandarizar los lenguajes de representación de reglas y los procesos de desarrollo. La motivación detrás de todo esto son los problemas de interoperabilidad y mapeo entre los productos comerciales y las aplicaciones de las organizaciones. Estos problemas han emergido de las fusiones y adquisiciones de las compañías, legislación industrial, regulaciones de los negocios y la necesidad de intercambiar información y conocimiento entre socios.

III. MODELADO CONCEPTUAL CON COMMONKADS

Nuestro enfoque para el diseño del SBC utiliza la metodología COMMONKADS (CK). En sus inicios, KADS se centraba en el problema del cuello de botella que suponía la adquisición de conocimiento, para posteriormente convertirse en una metodología completa para el desarrollo de SBC [8]. En la actualidad, COMMONKADS, nombre que recibe la evolución de KADS, cubre la gestión del proyecto, el análisis organizacional y los aspectos relativos a las Ingenierías del Software y Conocimiento relacionados con el desarrollo del SBC. En COMMONKADS se incorporan tres ideas que han emergido, de la experiencia en la Ingeniería del Conocimiento y la Ingeniería del Software. Estas tres ideas se pueden concretar en tres conceptos: modelado, reutilización y gestión del riesgo.

El principal producto que resulta de la aplicación de COMMONKADS es el conjunto de modelos. Este conjunto de modelos se puede considerar una agrupación estructurada de conocimiento que refleja todos aquellos aspectos importantes para que el SBC tenga éxito dentro de un contexto organizacional determinado. Para reflejar los diferentes aspectos del contexto en el cual se quiere implantar el SBC, COMMONKADS ofrece seis modelos [3]: organización, tareas, agentes, comunicación, conocimiento y diseño Figura 2. Todos estos modelos están relacionados entre sí y pueden ser configurados gracias a unas plantillas que la metodología ofrece para su confección. Los tres primeros modelos describen el contexto en el cual se va a desenvolver el SBC.

Los modelos de conocimiento y agentes proporcionan los requisitos de entrada que guiarán la implementación del sistema a través del modelo de diseño. Como se puede deducir, cada una de estas agrupaciones intentan responder a cada una de las preguntas claves en el desarrollo de un SBC: ¿Es un SBC la solución idónea para el problema que se quiere resolver?, pregunta que se puede responder por la descripción del contexto dado por los tres primeros modelos; ¿Cuál es la naturaleza y la estructura tanto del conocimiento como de la comunicación utilizada?, cuya respuesta se puede encontrar en el modelo de conocimiento y en el modelo de comunicación; y finalmente, ¿Cómo debe ser implementado el conocimiento?, pregunta que intenta esclarecer el modelo del diseño.

Se hace mención especial al Modelo del Conocimiento, que está descrito en [8], el cual describe la experticia que tiene un determinado agente y que es relevante para la consecución de una determinada tarea, además de describir la estructura del mismo en función de su uso. Obviamente, este modelo se hace sin hacer referencia a aspectos de implementación. Para poder llevar a cabo el modelado de los distintos papeles que puede jugar el conocimiento, éste está distribuido en tres categorías disjuntas:

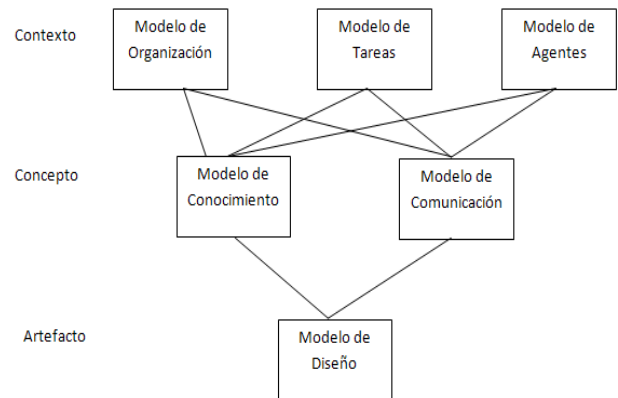


Figura 2: Modelos de la Metodología COMMONKADS [3]

- Conocimiento de tareas: Describe de una forma recursiva la descomposición de una tarea de alto nivel en varias subtareas. El conocimiento sobre una tarea se divide en dos partes: por un lado la tarea que sirve para especificar qué implica la aplicación de la misma, ya que define su objetivo en términos de los roles de entrada y de salida; por otro lado, está el método de la tarea, que define el

cómo se lleva a cabo la misma, indicando en qué subtareas se descompone y en qué orden deben de ser procesadas (control).

- **Conocimiento del dominio:** Especifica los hechos y asunciones que necesita el proceso de razonamiento para llevar a cabo su cometido en el dominio de la aplicación. El conocimiento puede ser estructurado en una serie de modelos del dominio que proporcionen una visión coherente de las distintas partes del mismo. Por lo tanto, en este apartado se va a especificar la forma, estructura y contenido del conocimiento relevante para la aplicación. La forma y la estructura constituyen lo que se denomina la ontología del dominio.
- **Conocimiento sobre inferencias:** Describe los procesos primitivos de razonamiento que tienen lugar en una aplicación, así como los roles de conocimiento que son usados por las inferencias. Obviamente, estos roles de conocimientos están relacionados con elementos del conocimiento del dominio. Hay que tener en cuenta, que las inferencias son consideradas primitivas respecto a un modelo de conocimiento determinado, ya que en otros modelos la misma inferencia puede ser una tarea descomponible.

COMMONKADS propone el lenguaje CML (*Conceptual Modelling Language*) [3], para materializar la especificación del modelo de conocimiento. Este lenguaje permite la definición de todos los elementos anteriormente citados, así como la estructuración del conocimiento en las partes anteriormente mencionadas. Además propone el uso de una notación gráfica, que permite no solo la definición de las estructuras de tareas (relación tareas-subtareas), sino que también permite la definición de la ontología, los conceptos del dominio y la definición de la dependencia de los datos entre las inferencias a través de las estructuras de inferencias. Aparte de CML, también se propone el uso de ML2 que está más orientado hacia la formalización del modelo. Esta descomposición del nivel de conocimiento en el conocimiento específico del dominio por un lado, y de las tareas y las inferencias por otro, favorece la reutilización del conocimiento en dos niveles. Al haber definido un conjunto de inferencias elementales independientes de la aplicación se permite que estas sean utilizadas en otros procesos de resolución de problemas.

El otro nivel de reutilización lo establece el conocimiento del dominio, que al ser definido independientemente de las tareas, puede ser reutilizado por otro conjunto de tareas definidas sobre el mismo dominio. Para favorecer la reutilización del conocimiento del dominio, COMMONKADS permite definir las ontologías a distintos niveles de generalización y distintos puntos de vista, lo que abre el abanico de posibles adaptaciones a otras aplicaciones. Otro de los aspectos importantes que introdujo COMMONKADS fue la definición de un marco de trabajo para la gestión y planificación del proyecto. CK define un ciclo de vida para el desarrollo del proyecto basado en un modelo en espiral como el propuesto por [9]. El modelo en espiral que plantea COMMONKADS se basa en los siguientes principios:

- La planificación del proyecto se centra principalmente en los productos y las salidas que tienen que producirse como resultado, más que un conjunto de actividades o fases.
- La planificación se realiza de una forma adaptativa a lo largo de una serie de ciclos en espiral, que están dirigidos por una valoración sistemática de los riesgos del proyecto.
- El control de calidad es una parte más de la gestión del proyecto, ya que la calidad está integrada en el desarrollo del SBC por medio de la metodología.

Estos principios están garantizados por un lado, por el conjunto de modelos, y por otro, por el ciclo de vida en espiral. Este ciclo de vida consta de cuatro fases:

- **Revisión:** Es el primer paso de cada ciclo y en él se revisa el estado actual del proyecto y se establecen los objetivos principales que se quieren cubrir en el ciclo en cuestión.
- **Valoración de riesgos:** Las líneas generales del proyecto establecidas en el paso anterior sirven de entradas para esta fase. Su función principal es la identificación y valoración de los principales obstáculos que nos podemos encontrar para la consecución exitosa del proyecto, así como las acciones que se deben tomar para minimizar dichos riesgos.
- **Planificación:** Una vez obtenida una visión clara de los objetivos que hay que cubrir, los riesgos que se pueden presentar y las acciones que hay que tomar, hay que realizar una planificación del trabajo a realizar. En dicha planificación, hay que establecer la distribución de la carga del trabajo en términos de qué tareas hay que realizar, una temporalización de dichas tareas, la distribución de los recursos, etc.
- **Monitorización:** Es la última fase del ciclo y está constituida por el desarrollo propiamente dicho. El trabajo realizado en esta fase está controlado y dirigido por el director del proyecto. Para determinar el grado de cumplimiento de los objetivos se requieren reuniones con los agentes implicados en el proyecto (usuarios, administradores, expertos,...). El resultado de dichas reuniones se utilizará como entrada del proceso de revisión del siguiente ciclo.

COMMONKADS abarca todo los aspectos del desarrollo de un SBC, proporcionando un marco de trabajo donde llevar a cabo la gestión del proyecto. Todo esto hace que COMMONKADS haya sido adoptado no solo en Europa, donde se ha convertido en un estándar de facto [3], sino que también empieza a ser utilizado en el resto del mundo.

No obstante, existe una necesidad de extender el uso de los lenguajes de modelado estandarizados de la ingeniería del software a la ingeniería del conocimiento, que permitan integrar, reusar e interoperar entre los diferentes sistemas y motores de inferencia [2]. Basado en lo anterior, se evidencia la importancia de proponer estrategias de trabajo que potencien la reutilización a un alto nivel de abstracción. La Arquitectura Dirigida por Modelos (MDA) se ha convertido en una iniciativa para propiciar soluciones bajo este contexto.

IV. ARQUITECTURA DIRIGIDA POR MODELOS (MDA)

La Arquitectura Dirigida por Modelos es un estándar del *Object Management Group* (OMG) que dirige completamente el ciclo de vida del diseño, despliegue, integración y manejo de las aplicaciones a través del uso de modelos en el desarrollo del software. MDA separa la especificación de la funcionalidad del sistema, de la especificación de la implementación sobre una plataforma de tecnología específica.

En consecuencia, MDA incentiva la especificación de un modelo independiente de la plataforma (PIM, en inglés) el cual no contiene información sobre la tecnología usada para llevar a cabo éste. Entonces, el PIM puede ser transformado a un modelo específico de la plataforma (PSM, en inglés) para incluir información acerca de la tecnología específica a utilizar para ser realizado. Posteriormente, cada PSM resultante puede ser transformado a código para ser ejecutado sobre la plataforma y obtener el producto final de software Figura 3. Sobre el tope de estos modelos, MDA también presenta un modelo computacional independiente (CIM, en inglés). Éste describe el sistema dentro de su medio ambiente (por ejemplo: dominio del negocio) y muestra lo que se espera haga el sistema sin enseñar detalles acerca de cómo será construido.

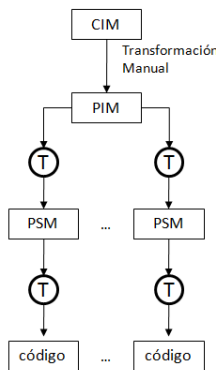


Figura 3: Marco de Trabajo de Arquitectura Dirigida por Modelos [10]

El CIM, PIM y el PSM pueden ser desarrollados usando cualquier lenguaje de especificación, pero típicamente es usado UML desde que éste es un lenguaje de modelado estándar para propósito general y al mismo tiempo puede ser extendido para definir lenguajes especializados para ciertos dominios (por ejemplo: extensión de metamodelos o perfiles).

V. MARCO DE REFERENCIA METODOLÓGICO PROPUESTO

Para el diseño del SBC en esta investigación, se propone la aplicación del marco de referencia metodológico basado en MDA de la Figura 4, obtenido de realizar una combinación entre COMMONKADS y la Arquitectura Dirigida por Modelos.

La selección de los distintos artefactos escogidos para esta propuesta, están fundamentados en unificar el lenguaje de modelado del conocimiento y proveer de transformaciones que permitan el mapeo o trazabilidad entre el diseño y la implementación del sistema.

De la Figura 4, observamos un:

A. Modelo Independiente de la Computación (CIM)

Compuesto por un diagrama de casos de uso y los 3 primeros modelos de COMMONKADS. Su objetivo al haber sido incluidos, es tener una visión global del negocio o la organización, sus procesos, tareas y recursos, que servirán de guía para generar el PIM. Para cada modelo perteneciente al CIM, se seleccionan y establecen los siguientes artefactos provenientes de COMMONKADS:

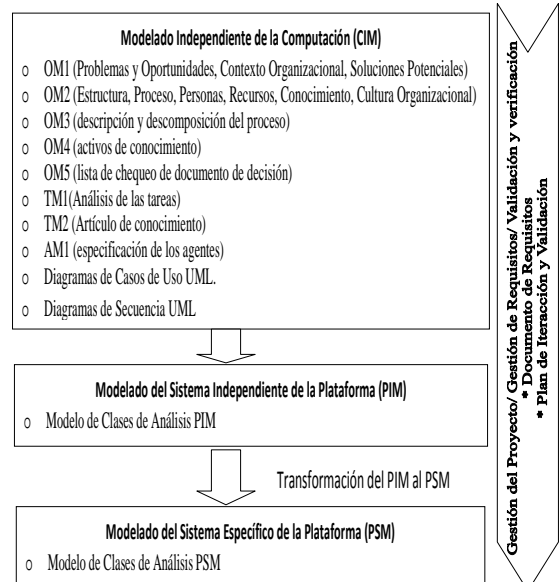


Figura 4: Marco de Referencia Metodológico Propuesto Basado en MDA para el Diseño de un SBC

Modelo de Organización:

- OM1 (problemas y oportunidades, contexto organizacional, soluciones potenciales)
- OM2 (estructura, proceso, personas, recursos, conocimiento, cultura organizacional)
- OM3 (descripción y descomposición del proceso, diagramas de actividad)
- OM4 (activos de conocimiento)
- OM5 (lista de chequeo de documento de decisión)

Modelo de Tareas:

- TM1 (análisis de las tareas)
- TM2 (artículo de conocimiento)

Modelo de Agentes:

- AM1 (especificación de los agentes)

Así mismo, de la psicología cognitiva logramos obtener, que el ser humano puede dividir sus procesos intensivos de conocimiento, en tareas de razonamiento [8]:

- Analíticas: por lo general el sistema a estudiar “pre-existe”, pero típicamente no es completamente conocido. Las tareas analíticas se pueden dividir en: clasificación, evaluación, diagnóstico, predicción, monitoreo.
- De síntesis: en estas tareas, el sistema por lo general no existe y debe ser construido. Las mismas se pueden clasificar en: síntesis, diseño, asignación, planificación, calendarización y modelación.

De esta manera, se representan a nivel general las distintas actividades que podría realizar un SBC. El mismo es diagramado en UML y puede ser visto en la Figura 5.

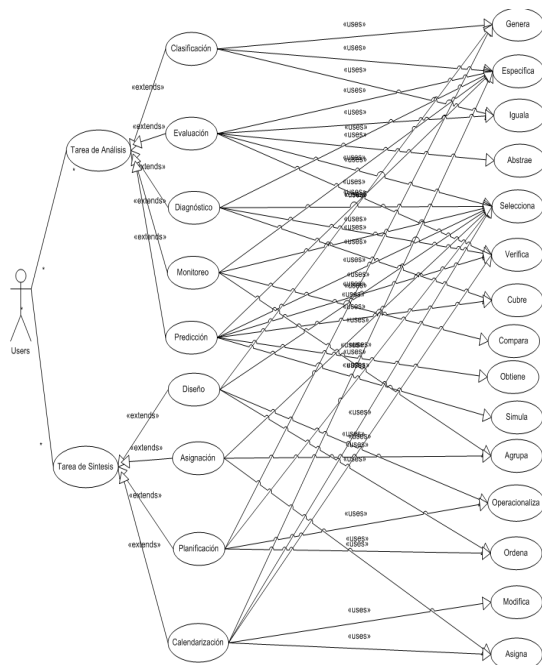


Figura 5: Diagrama de Casos de Uso General para el CIM

B. Modelo del Sistema Independiente de la Plataforma (PIM)

Para obtener el modelo independiente de la plataforma, se selecciona COMMONKADS para extraer de éste los artefactos y conceptos que nos ayudarán a la creación del modelo. Entre los conceptos más importantes identificados se mencionan:

- Tareas: representan funciones de razonamiento complejo. Una tarea define una meta de razonamiento en términos de una entrada y una salida.
- Métodos de Tarea: describe cómo una tarea puede ser realizada a través de la descomposición de sub-funciones.
- Inferencias: describe un paso de razonamiento primitivo, o lo que es igual a decir, una actividad de razonamiento simple. La realización de varias inferencias puede dar lugar al cumplimiento de una tarea.
- Roles de conocimiento: son nombres abstractos de objetos de datos que tienen una participación en el proceso de razonamiento. Por ejemplo: “hipótesis”, es un nombre

funcional para un objeto en el dominio que juega el rol de una solución candidata.

- Dominio del conocimiento: describe la información estática y los objetos del conocimiento para el dominio de la aplicación. Un dominio de conocimiento típicamente está compuesto por un esquema de dominio y una base de conocimiento.

Basándonos en todo lo anterior, se establecen un conjunto de diagramas de secuencia UML y un modelo conceptual genérico, que posteriormente podrán ser instanciados dependiendo de la tarea a realizar por el SBC, los cuales pueden ser apreciados en las Figuras 6, 7, 8, no siendo mostrados todos los artefactos por limitaciones de espacio.

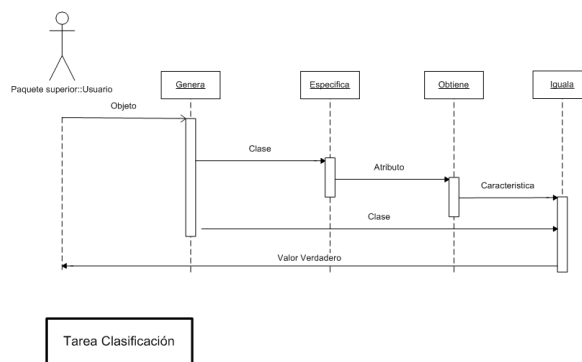


Figura 6: Diagrama de Secuencia Genérico “Clasificación” para el PIM

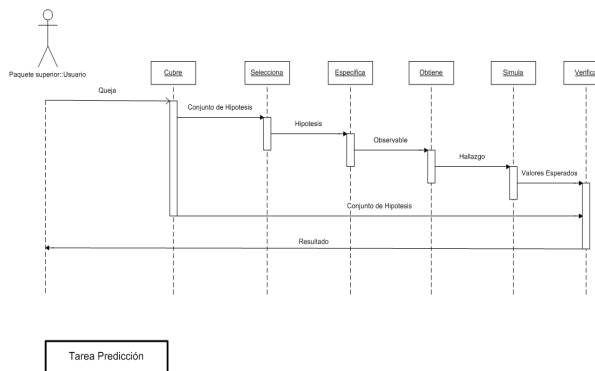


Figura 7: Diagrama de Secuencia Genérico “Predicción” para el PIM

C. Modelo del Sistema Específico de la Plataforma (PSM)

Conformado por un diagrama de clases que modela los componentes de un SBC a nivel general para la plataforma Java. Éste permitirá posteriormente instanciar los distintos elementos resultantes al modelo específico de la plataforma para un caso de estudio. El mismo, puede ser visto en la Figura 9.

Se escoge Java, debido a sus diversas ventajas, entre ellas: su soporte para múltiples plataformas, el respaldo de empresas reconocidas, arquitectura de varias capas y atributos de calidad.

TRANSFORMACIÓN: representa el mapeo o trazabilidad que permitirá convertir el PIM en un PSM. Existen diferentes enfoques para dar a un modelo la información necesaria. La OMG propone 4 enfoques para realizar esta transformación [10]. Estas son:

- Transformación Manual
- Transformación de un PIM preparado utilizando un perfil.
- Transformación usando patrones y marcas
- Transformación automática

Para este caso, fue utilizada la transformación manual, obteniendo el Mapeo de la Tabla I y Figura 10.

Tabla I: Mapeo PIM a PSM

PIM	Transformación	PSM
Tarea	T1	Class
Método de Tarea	T2	Method
Inferencia	T3	Method
Dominio de Conocimiento	T4	Package
Base de Conocimiento	T5	Method
Concepto	T6	Class
Función de Transferencia	T7	Method
Relaciones	T8	Class
Atributo de Valor	T9	Attribute
Roles de Conocimiento	T10	Method
Tipo de Reglas	T11	Method
Esquema de Dominio	T12	Class

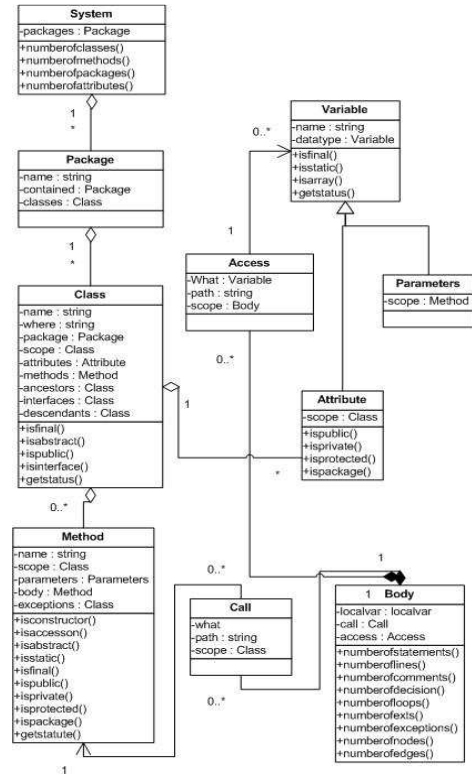


Figura 9: Diagrama de Clases General para el PSM

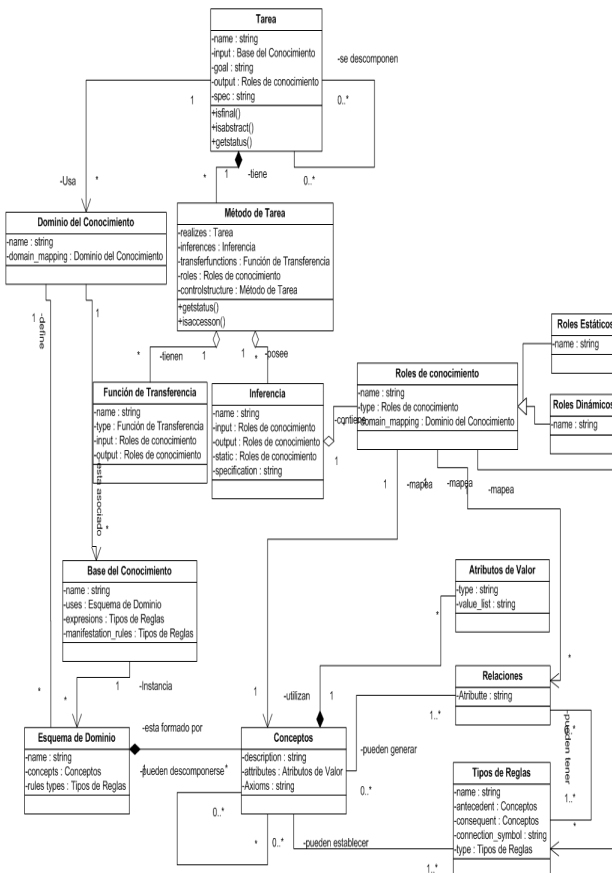


Figura 8: Modelo Conceptual General para el PIM

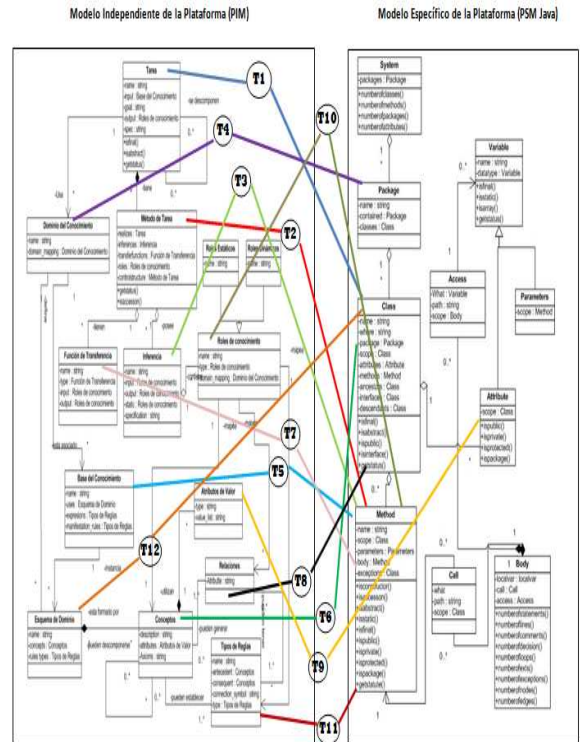


Figura 10: Transformaciones PIM a PSM

VI. INSTANCIACIÓN DEL MARCO DE REFERENCIA
 METODOLÓGICO

Después de haber definido el PIM, PSM y las transformaciones en el apartado anterior, se procede a instanciar los mismos. Todos estos orientados al marco de referencia mostrado en la Figura 4, para el modelado de un sistema que permita el pronóstico del cáncer de seno.

El proceso de gestión del paciente con cáncer de seno sigue generalmente el siguiente flujo de tareas: diagnóstico inicial, confirmación del diagnóstico, pronóstico, asignación de la terapia y monitoreo. El pronóstico como la predicción del comportamiento del tumor, es un punto crucial en la gestión del paciente. La definición de esta tarea depende principalmente si estamos buscando propiedades actuales del cáncer de seno o si deseamos identificar estados actuales no observables (por ejemplo, pasado, presente y futuro de características de crecimiento biológico)

Bajo este orden de ideas y revisando la definición de pronóstico en [11], indica que estamos tratando a nivel cognoscitivo con un problema de “análisis” con el objetivo de “predecir” un comportamiento, el cual según la Figura 5, involucra las sub-tareas de cubrir, seleccionar, especificar, obtener, simular y verificar. Basándonos en estas premisas, se procede a aplicar el marco metodológico propuesto.

A. Obteniendo El CIM Instanciado (Paso 1)

Para conseguir el CIM, se realizan todos los artefactos planteados en la sección V para el modelo organizacional, de tareas y agentes. Sin embargo, no son presentados completamente todos los artefactos por limitaciones de espacio. A continuación se muestran algunos en las Tablas II, III, IV:

Modelo Organizacional

Tabla II: Artefacto OM3 (Modelo Organizacional)

#	Tarea	Ejecutada por	Donde	Activo de Conocimiento	Intensivo	Significancia
1	Diagnóstico Inicial	Imagenólogo/ Bioanalista	Centro de Diagnóstico de Imagen	Clasificación Internacional de Enfermedades con modificación clínica	Sí	2
2	Confirmación del Diagnóstico	Médico Oncólogo/ Junta Médica	Hospital o Clínica	* Diagnóstico y Tratamiento de cáncer de seno	Sí	3
3	Pronóstico	Médico Oncólogo	Hospital o Clínica	* Guía sobre los factores de pronóstico en el cáncer de seno	Sí	4
4	Selección de la Terapia	Médico Oncólogo/ Junta Médica	Hospital o Clínica	* Guía para el tratamiento multidisciplinario en el cáncer de seno operable * Guía para Quimioterapia primaria en el cáncer de seno operable	Sí	4
5	Seguimiento	Médico Oncólogo	Hospital o Clínica	* Guía para el tratamiento multidisciplinario en el cáncer de seno operable	Sí	3

1= Menos Significativo
 5= Muy Significativo

Modelo de Tareas

Tabla III: Artefacto TM2 (Modelo de Tareas)

Nombre	Guía sobre los factores de pronóstico en el cáncer de seno	
Poseído por	Médico Oncólogo / Sistema de Conocimiento	
Usado en	3.- Pronóstico	
Dominio	Cáncer de Seno	
Naturaleza del conocimiento		Cuello de botella / Para ser mejorado
Formal, riguroso	X	
Empírico, cuantitativo		
Heurístico, reglas de oro		
Altamente especializado, específico del dominio	X	
Basado en la experiencia	X	
Basado en la acción		
Incompleto		
Incierto, podría ser incorrecto		
Rápidamente cambiante	X	X
Difícil de verificar		
Tácito, difícil de transferir		
Forma del conocimiento		
En la mente		
Papel	X	
Electrónico	X	
Habilidad de acción		
Otro		
Disponibilidad del conocimiento		
Limitaciones en tiempo	X	
Limitaciones en espacio		
Limitaciones en acceso		
Limitaciones en calidad		
Limitaciones en forma	X	X

Modelo de Agente

Tabla IV: Artefacto AMI (Modelo de Agente)

Nombre	Médico Oncólogo
Organización	Hospital o Clínica. Juega el rol primario en el diagnóstico y pronóstico del paciente
Involucrado en	2.- Confirmación del diagnóstico 3.- Pronóstico 4.- Selección de la terapia
Se comunica con	5.- Seguimiento Base de Datos Expertos / Investigadores Junta Médica Pacientes
Conocimiento	Diagnóstico y tratamiento del cáncer de seno Factores de pronóstico en el cáncer de seno Tratamiento para el cáncer de seno Quimioterapia para el cáncer de seno
Otras competencias	Habilidad para investigar, evaluar y discernir entre distintos criterios de expertos Asegurarse de la selección de una terapia adecuada que maximice las posibilidades favorables de curación de la enfermedad, basado en un correcto diagnóstico y
Responsabilidades y limitaciones	pronóstico de la enfermedad del paciente

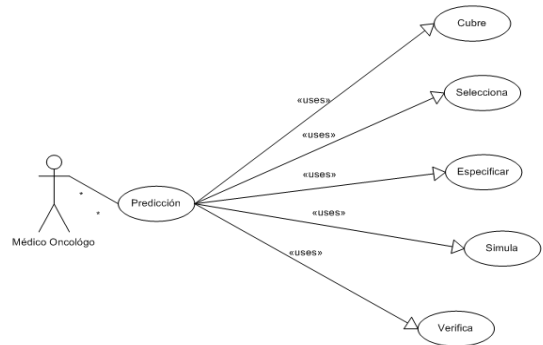


Figura 11: Transformaciones PIM a PSM

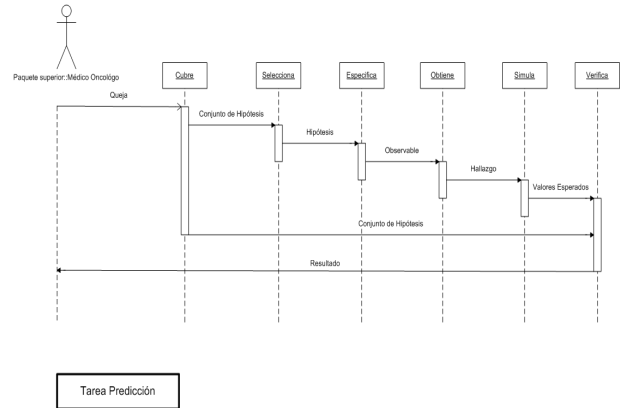


Figura 12: Diagrama de Secuencia para el "Pronóstico" del Cáncer de Seno

También es generado el diagrama de casos de uso, que resulta de seleccionar la tarea analítica "predicción" del diagrama general que se expuso en la Figura 5. El mismo puede ser apreciado en la Figura 11.

B. Obteniendo El PIM Instanciado (Paso 2)

Para obtener el PIM como se explicó previamente, se selecciona del conjunto de diagramas de secuencia mostrados en la sección V, aquella tarea que más se adapte a la actividad de conocimiento a realizar por el SBC, en nuestro caso "predicción", el cual puede ser apreciado en la Figura 12.

Adicionalmente, basados en el modelo conceptual de la Figura 8 y los factores de pronóstico de cáncer de seno [11], se obtiene el modelo conceptual del SBC para el pronóstico del cáncer de seno mostrado en la Figura 13.

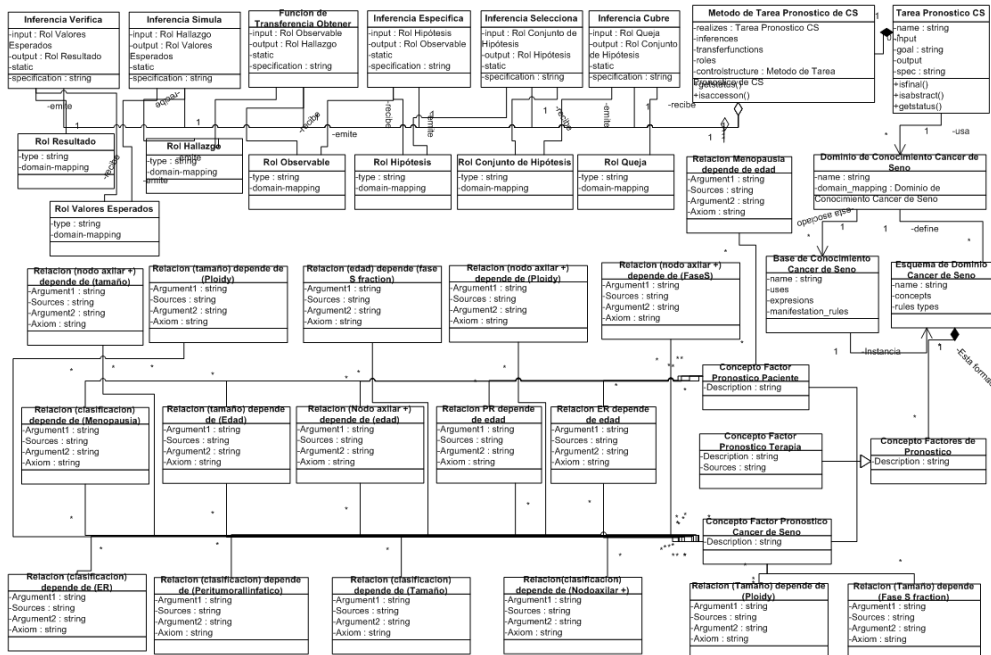


Figura 13: Modelo Conceptual del SBC para el "Pronóstico" del Cáncer de Seno

C. Obteniendo El PSM Instanciado (Paso 3)

Aplicando las transformaciones presentadas en la Figura 10 al modelo conceptual de la Figura 13 y verificando el resultado con el diagrama de clases de la Figura 9, se obtiene el modelo específico de la plataforma del Sistema Basado en Conocimiento para el pronóstico del cáncer de seno. El mismo puede ser apreciado en la Figura 14.

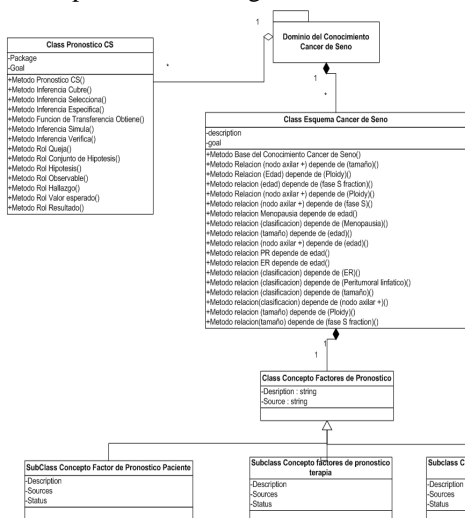


Figura 14: Diagrama de Clases del SBC para el “Pronóstico” del Cáncer de Seno

VII. CONCLUSIONES

Este trabajo ha mostrado la importancia de los sistemas basados en conocimiento y su problemática en la carencia de metodologías que establezcan un mapeo efectivo entre el modelo conceptual y el modelo de diseño.

Particularmente, se estudió COMMONKADS, de la cual se pudo observar que aunque es bastante completa y abarca la mayoría de las fases del ciclo de vida de estos sistemas, adolece de un lenguaje de representación estándar para su producto clave “el conocimiento”. Lo anterior, aunado a la ausencia de una trazabilidad clara entre sus modelos conceptuales y de diseño, genera una limitante para el desarrollo, reutilización e interoperabilidad de los mismos.

Con esta investigación, se plantea una alternativa a través de MDA buscando disminuir las carencias citadas anteriormente. El marco de referencia propuesto consigue establecer una trazabilidad entre el modelo conceptual y de diseño, así como unificar el lenguaje de modelado a través de UML, el cual es

ampliamente reconocido hoy en día por los ingenieros y desarrolladores de software, lo que permite disminuir los costos en los tiempos de aprendizaje y ganar mayor interoperabilidad entre los diversos sistemas.

Mediante el caso de estudio, se logró validar el marco de referencia propuesto, obteniendo tiempos de modelado considerablemente menores en comparación con lo que toma aplicar la metodología pura y simple, además de aprender la misma. No obstante, como resultado de nuestra experiencia, logramos validar que también se pierde un poco la riqueza semántica del modelo de conocimiento, por lo cual recomendamos la combinación de estas metodologías ya existentes con la propuesta planteada, para obtener un mejor modelado del sistema.

Se propone como trabajo futuro hacer una extensión o perfil para UML dirigido a los SBC, para evitar la pérdida del contenido semántico en el modelado.

REFERENCIAS

- [1] P. Drucker. *Los Desafíos de la Gerencia para el El Siglo XXI*. 3th Edition, Norma, Febrero 1999.
- [2] J. Cañadas, S. Túnez, J. Palma, *Aplicación de MDA a la Ingeniería del Conocimiento*, unpublished.
- [3] J. Palma, E. Paniagua, F. Martin, R. Marin, *Ingeniería del Conocimiento. de la Extracción al Modelado del Conocimiento*, Revista Iberoamericana de Inteligencia Artificial, vol. 4, no. 11, pp. 46-72, Abril 2000.
- [4] M. del Águila, J. Cañadas, J. Palma, S. Túnez, *Towards a Methodology for Hybrid Systems Software Development*, in proceeding of the XVIII International Conference on Software Engineering & Knowledge Engineering (SEKE 06), San Francisco, USA, July 2006.
- [5] M. Abdullah, *A UML Profile for Conceptual Modeling of Knowledge Base System*, Ph.D. thesis, Department of Computer Science, University of York, York, United Kingdom, April 2006.
- [6] *KBE Services for Engineering Design*, <http://www.omg.org/schedule/>.
- [7] M. Henao, *COMMONKADS-RT: Una Metodología para el Desarrollo de Sistemas Basados en el Conocimiento de Tiempo Real*, thesis Doctoral, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, España, Abril 2001.
- [8] A. Schreiber, J. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van del Velde, J. Wielinga, *Knowledge Engineering and Management. The COMMONKADS Methodology*, 1st Edition, MIT Press, February, 2000.
- [9] B. Boehm, *A Spiral Model of Software Development and Enhancement*, IEEE Computer, vol. 21, no. 5, pp. 61-72, May 1988.
- [10] *MDA Guide Version 1.0.1*, <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [11] S. Roberto. *Using COMMONKADS to Build an Expertise Model for Breast Cancer Prognosis and Therapy*. RFC 2737, November 1995.