

UNA VISIÓN COGNITIVA DE MODELOS DE REQUISITOS DE SOFTWARE

Gladys Kaplan⁽²⁾ *Jorge H. Doorn*⁽¹⁾⁽²⁾ *Graciela Hadad*⁽²⁾

(1) INTIA, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

(2) Universidad Nacional de La Matanza, Argentina

Tel/Fax: 54-2293-439681

e-mail: {gladyskaplan, gracielahadad} @gmail.com, jdoorn @exa.unicen.edu.ar

RESUMEN

La Ingeniería de Requisitos provee métodos, técnicas y herramientas para ayudar a los ingenieros a elicitar y especificar requisitos. Uno de los primeros modelos que se construye en muchos procesos de requisitos es un glosario. Por otra parte el uso de glosarios se ha constituido en los últimos años, en una práctica común en otras actividades del desarrollo de software. En este proyecto se aspira a aplicar visiones de la Gestión del Conocimiento en general y de las Ontologías en particular a un modelo específico de glosario: el Léxico Extendido del Lenguaje con el fin mejorar, en algún sentido su construcción, pero fundamentalmente su tratamiento como reservorio de información.

INTRODUCCIÓN

Ha sido reiteradamente comprobado que en todo proceso de definición y / o elicitación de los requisitos de un sistema de software, debe conocerse el contexto en el que dicho sistema deberá funcionar [Anton96].

Casi naturalmente la actividad de elicitar conocimiento acerca de este contexto y su modelado posterior utilizando alguna técnica de representación se realiza como primera etapa del ciclo de trabajo [Sommerville02] [Coulin05][Hull05][Young04].

Para predecir como será el proceso del negocio o contexto organizacional en el que se desempeñará el sistema de software, es muy conveniente conocer con anterioridad el proceso del negocio existente en el momento de comenzar a estudiar su posible implantación [Parviainen05]. En otras palabras la Ingeniería de Requisitos debe lidiar tanto con el proceso del negocio actual como con los planes, proyectos y expectativas que se tienen respecto de cómo será el proceso del negocio una vez que el sistema de software proyectado se ponga en servicio.

Los modelos que se construyan para representar el proceso del negocio previo al sistema tienen la característica de ser modelos descriptivos, mientras que los que se relacionan con el contexto del sistema de software serán esencialmente prescriptivos [Seidewitz03] [Pressman06] [Hull05].

Independientemente de la técnica de modelado que se utilice el ingeniero de requisitos se encuentra con actividades, situaciones, actores, recursos, que forman parte del proceso de negocio que esta siendo estudiado.

Los dos aspectos que más obscurecen la comprensión del proceso del negocio son, por un lado la compleja maraña de relaciones existentes entre estas actividades, situaciones, actores y recursos y por el otro el importante grado de variabilidad de las mismas [Dietz06].

La primera impresión que le surge a un observador poco entrenado es la existencia de gran diversidad y complejidad de relaciones acompañada en mayor o menor grado por una pobre o carente estructura y una falta de lógica en las mismas [Dietz06].

Una pobre comprensión de las razones subyacentes en el proceso del negocio enturbia aún más la percepción y razones de las evoluciones que en el mismo se producen.

Desde el punto de vista de los requisitos de un sistema de software y recordando que la construcción del mismo no es instantánea, resulta evidente que los dos aspectos mencionados son notoriamente perturbadores, el primero porque se puede comenzar un desarrollo sobre una base poco sólida y el segundo porque la realidad sorprenderá al grupo de desarrollo de software con modificaciones inesperadas en el proceso del negocio o peor aún poniendo en evidencia que los cambios previstos en el proceso del negocio no tendrán lugar y sí ocurrirán otros no previstos.

La Ingeniería de Requisitos a lo largo de toda su historia ha desarrollado gran cantidad de estrategias con mayor o menor éxito para atemperar este tipo de inconvenientes.

Desde un punto de vista algo diferente, en los estudios de Gestión del Conocimiento en organizaciones en general y de las ontologías en particular se han desarrollado una cantidad importante de estrategias para extraer y modelar conocimiento esencial de diferentes dominios.

En el ámbito de la Gestión del Conocimiento no es relevante el grado de formalismo logrado en la ontología que se desarrolle [Calero06] y su capacidad de producir inferencias o habilidad “informa” [Dietz06] sino la capacidad de transmitir conocimiento a otros seres humanos.

En algún sentido los enfoques basados en la Gestión de Conocimiento y en las ontologías han desarrollado visiones más profundas o esenciales del dominio bajo estudio, sin embargo, en el dominio particular de la definición y gerenciamiento de los requisitos de un sistema de software no se ha avanzado mucho más que en algunos principios generales y algunos metamodelos de algún proceso de requisitos. Por otro lado resulta evidente que la riqueza, el detalle y la calidad de muchos procesos de requisitos propuestos y / o en práctica industrial actual, superan notoriamente los procesos conocidos de desarrollo de ontologías.

En el presente artículo se propone aplicar las estrategias básicas de las ontologías utilizadas en la Gestión del Conocimiento para evaluar, leer en forma más profunda y enriquecer un proceso de requisitos particular suficientemente afianzado como para dar soporte a un estudio de esta naturaleza.

La sección 2 está dedicada a describir brevemente el proceso de requisitos en el que se enmarca este artículo, la sección 3 describe el Léxico Extendido del Lenguaje (LEL) y la sección 4 presenta una breve descripción de las Ontologías y como se encuadra el LEL en las mismas.

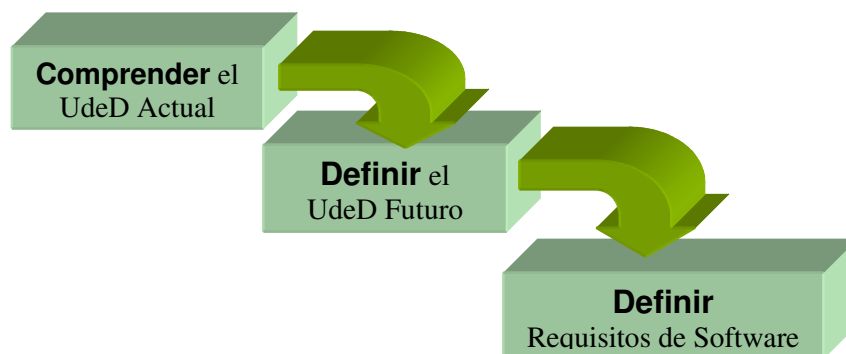


Figura 1- Estrategia para capturar requisitos de software

PROCESO DE REQUISITOS

La estrategia en la que se enmarca el presente artículo [Leite05], propone modelar las situaciones del proceso del negocio actual y de aquellas previstas cuando el nuevo software se ponga en servicio, en

lenguaje natural, produciendo los requisitos del sistema de software, también en lenguaje natural. Esta estrategia comienza (ver Figura 1) con la comprensión del UdeD ¹ (UdeD) actual, pasando luego a definir el UdeD futuro para finalmente definir los requisitos del software.

La etapa **Comprensión del UdeD Actual**, se sustenta en la construcción y uso de dos modelos basados en Lenguaje Natural: el Léxico Extendido del Lenguaje (LEL) [Leite90], un glosario levemente formalizado del vocabulario del dominio de la aplicación, y Escenarios Actuales (EA), los que contienen descripciones de situaciones en el UdeD. El principal propósito del léxico es la captura del vocabulario del dominio de la aplicación, mientras que la comprensión de la funcionalidad y características del UdeD se obtienen por medio de los escenarios [Leite00]. Cada escenario se describe usando los términos definidos en el LEL.

Es justamente en esta primera etapa, en la que la confluencia de las estrategias de la Gestión del Conocimiento y la gran base de experiencias de la Ingeniería de Requisitos pueden amalgamarse produciendo un proceso de requisitos con más énfasis en los aspectos esenciales del proceso del negocio.

Adicionalmente, toda información relevante pero no representable en estos modelos se registra en una ficha de información extemporánea [Kaplan06], para ser utilizada en etapas posteriores del proceso.

La etapa **Comprensión del UdeD Futuro**, se lleva a cabo construyendo Escenarios Futuros (EF) [Doorn02] los que tienen la misma representación que los Escenarios Actuales pero prescriben situaciones futuras o planificadas acerca de cómo será el proceso del negocio cuando se ponga en servicio el sistema de software que se está empezando a desarrollar.

Finalmente, durante la etapa de Definición de los Requisitos de Software se extraen las características que deberá satisfacer el sistema de software a partir del estudio de los roles que para él se previeron en los Escenarios Futuros.

LEXICO EXTENDIDO DEL LENGUAJE

Durante la construcción del LEL [Hadad06] se registran símbolos (palabras o frases) peculiares o relevantes del dominio de la aplicación. Según [Jackson95] “El dominio de la aplicación no se limita solo a las partes del mundo que están directamente conectados con la maquina. Es decir, el dominio de la aplicación es todo lo concerniente al sistema de información que se está analizando y donde coexisten los requisitos del software. Todo lo que es importante para los requisitos aparece de alguna manera en el dominio de la aplicación.”

Es importante resaltar que durante esta etapa el ingeniero de requisitos debe centrarse en **conocer el vocabulario de la aplicación** dejando para una etapa posterior la comprensión resto del problema.

Cada símbolo se identifica con un nombre (o más de uno en caso de sinónimos) y su descripción consta de dos partes. Una llamada Noción que describe la denotación del símbolo y la otra Impacto que describe la connotación del mismo (ver Figura 2). Los símbolos se clasifican en cuatro tipos de acuerdo a su uso general en el UdeD. Estos tipos son: Sujeto, Objeto, Verbo y Estado.

Al describir los símbolos, dos principios deben ser respetados: el principio de circularidad que postula la maximización del uso de símbolos en la descripción de otros símbolos y el principio de

¹ Contexto general en el cual el software deberá ser desarrollado y deberá operar. Incluye todas las fuentes de información y todas las personas relacionadas con el mismo.

vocabulario mínimo que postula la minimización del uso de términos que externos al léxico. Estos términos externos deben pertenecer a un pequeño subconjunto de un diccionario en lenguaje natural. Estas reglas enfatizan la descripción del vocabulario como un hipertexto auto contenido y altamente conectado.

El LEL es un documento intencionadamente creado de tal manera que la descripción de cada símbolo sea altamente dependiente del contexto, esto se debe a que se busca enfatizar el uso de los símbolos en desmedro de vocabulario externo (principio de circularidad). Luego, como toda estructura endógena es altamente sensible a la salud de su propia estructura. La falta de claridad o la falta de información acerca de un símbolo tienen un impacto mucho más alto que una documentación no endógena.

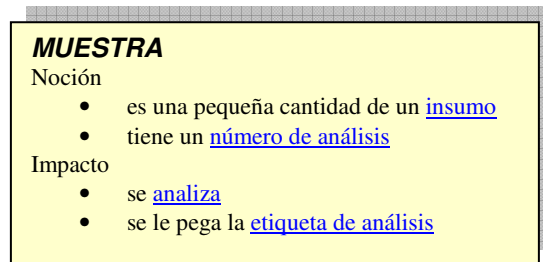


Figura 2 - Ejemplo de un símbolo del LEL tipo Objeto

LA VISIÓN DE LA GESTIÓN DEL CONOCIMIENTO

Basándose en la riqueza de la estructura interna Ruíz e Hilera [Ruíz 06] clasifica las ontologías en las siguientes categorías:

Vocabularios controlados: Formados por una lista de términos.

Glosarios: Listas de términos con sus definiciones en lenguaje natural.

Tesauros: agregando a los anteriores sinónimos.

Jerarquías informales: Jerarquías que no siguen una taxonomía estricta.

Jerarquías formales: Jerarquías con uso estricto de la relación es-un.

Marcos: Jerarquías que incluyen la herencia de propiedades.

Ontologías con restricciones de valor: Los tipos de datos de las propiedades pueden tener restricciones.

Ontologías con restricciones lógicas: Permitiendo la existencia de reglas lógicas entre los términos.

Una simplificación de esta clasificación refiere a las ontologías como livianas o pesadas, siendo las primeras las más livianas y las últimas las más pesadas.

Observando la estructura de LEL se tiene que obviamente la misma es más rica que la de los Vocabularios Controlados, que la de los Glosarios y los Tesauros. Al arribar la comparación con las ontologías en las que los términos están organizados en jerarquías se observa que el LEL contiene jerarquías y que las mismas son estrictas aunque adolece de dos debilidades que ameritan ser analizadas. Estas debilidades son: i) no todos los símbolos del LEL pertenecen a una jerarquía y ii) la representación de la jerarquía es menos clara que la utilizada en la mayoría de las ontologías existentes.

En este punto empiezan a abundar las preguntas y a faltar las respuestas. Es entonces justamente aquí donde se plantean las principales hipótesis de trabajo que guían el presente proyecto.

Por un lado no se sabe hasta que punto la existencia de taxonomías incompletas es una debilidad ya que del estudio de algunos casos prácticos parece surgir que existe información y aparentemente valiosa en las omisiones de componentes de una jerarquía. Retornando a la semántica del LEL y de las

ontologías se tiene que el primero aspira a describir el lenguaje utilizado en un cierto contexto, mientras que una ontología pretende describir todo el lenguaje relacionado con un cierto contexto sin considerar el grado de uso y de internalización de los términos en los actores del UdeD. Desde este punto de vista parecen ser muy relevantes las omisiones en las jerarquías del LEL.

Por otro lado, la representación de las jerarquías como parte de las nociones de los símbolos, es un asunto fundamentalmente sintáctico y posiblemente se pueda usar las mismas sin mayores inconvenientes, una vez que los usuarios tengan en cuenta esta característica. Esto deberá ser verificado con casos de estudio y ejercitación de su uso.

De todo lo anterior se puede deducir que el LEL se debiera ubicar en la clasificación de Ruíz e Hilera en algún lugar cercano a las Jerarquías formales. Sin embargo esta visión es claramente incompleta ya que hasta el momento se ha ignorado la importancia de los vínculos entre los símbolos del LEL, los cuales relacionan de una manera muy rica miembros de una jerarquía con miembros de otra con contenido semántico muy variado.

Aquí se hace evidente que la clasificación de Ruíz e Hilera crea una suerte de “continuum” como ellos mismos definen entre las ontologías livianas y las ontologías pesadas, por la incorporación de herencia, restricciones de valores y relaciones lógicas a las jerarquías formales. La riqueza semántica del LEL se aparta también de las jerarquías formales pero en un plano notoriamente diferenciado.

Este hecho resalta la última pregunta que surge de observar el LEL desde la visión de la Gestión del Conocimiento, que consiste básicamente intentar conocer cuan valiosa es para la Ingeniería de Requisitos esta red de referencias internas del LEL.

Muchos y muy variados experimentos se pueden hacer en relación con este último punto. El primero de ellos, que se encuentra en su fase inicial, consiste en tomar un ejemplo de un caso real de un LEL correctamente construido y utilizar el mismo para construir una jerarquía formal. A partir de esta actividad se planifica registrar tres conjuntos de información:

Información no disponible en el LEL necesaria para construir la jerarquía formal.

Información presente en el LEL no registrable en la jerarquía formal

Información trasladable de un modelo al otro.

Cada uno de estos conjuntos será analizado desde el punto de vista de los requisitos del sistema de software previamente obtenidos para ese caso. Se asume que los resultados de este tipo de experimentos lograrán estimar la importancia relativa de las debilidades y fortalezas del LEL como herramienta utilizada en la ingeniería de requisitos.

CONCLUSIONES

Se han integrado parcialmente los marcos teóricos de las ontologías livianas e intermedias con los correspondientes a las primeras etapas de la Ingeniería de Requisitos.

Se han detectado las principales diferencias de ambos marcos referenciales y se han precisado las fuentes de riesgos y de oportunidades que surgen de dicha integración.

Se ha diseñado un primer experimento de comparación de los efectos prácticos de las diferencias de ambas estrategias.

Se planifica llevar a cabo el experimento concebido, reformulando y profundizando el proyecto de investigación de acuerdo a los resultados que se obtengan.

REFERENCIAS

- [Anton96] Anton Annie, "Goal Based Requirements Analysis," in Proc. Second Int. Conference on Requirements Engineering., ICRE '96, pp. 136–144, 1996.
- [Calero06] Calero, C., Ruiz, F., Piattini, M., Preface, in "Ontologies for Software Engineering and Software Technology", Calero, C., Ruiz, F., Piattini, M., Preface, Springer, New York, 2006, ISBN 978-3-540-34517-6.
- [Coulin05] Coulin, C., Zowghi, D., "Requirements Elicitation for Complex Systems: Theory and Practice", in Requirements Engineering for Sociotechnical Systems, J.L. Maté & A. Silva (eds), Chapter III, pp37-52, 2005, ISBN 1-59140-506-8.
- [Dietz06] Dietz, J.L.G., "Enterprise Ontology. Theory and Methodology", Springer, New York, ISBN 978-3-540-26169-5, 2006.
- [Doorn02] Doorn J.H., Hadad G.D.S., Kaplan G.N., "Comprendiendo el Universo de Discurso Futuro con Escenarios", Workshop Requirements Engineering 02, pp 117- 131, 2002
- [Garda05] Garda, J.A., Casal, J.A., Garcia Vasquez, R., Rodriguez Yáñez, S., "Conceptual Modeling in Requirements: weaknesses and Alternatives", in Requirements Engineering for Sociotechnical Systems, J.L. Maté & A. Silva, Chapter III, pp53-67, 2005 , ISBN 1-59140-506-8.
- [Hadad06] Hadad, G.D.S., Doorn J.H., Kaplan G.N., "Creating Software System Context Glossaries", in Encyclopedia of Information Science and Technology, Second edition, Mehdi Khosrow-Pour (ed.), Idea Publishing, fecha estimada de publicación marzo 2009.
- [Hull05] Hull, E., Jackson, K., Dick, J., "Requirements Engineering", Second Edition, Springer, 2005, ISBN 1-85233-879-2.
- [Jackson95] Jackson, M., "Software Requirements & Specifications. A lexicon of practice, principles and prejudices", Addison Wesley, ACM Press, 1995.
- [Kaplan06] Kaplan G.N., Doorn J.H., Hadad, G.D.S., "Handling Extemporaneous Information in Requirements Engineering", in Encyclopedia of Information Science and Technology, Second edition, Mehdi Khosrow-Pour (ed.), Idea Publishing, fecha estimada de publicación marzo 2009.
- [Leite90] Leite, J.C.S.P., Franco, A.P.M., "O Uso de Hipertexto na Elicitação de Linguagens da Aplicação", Anais de IV Simpósio Brasileiro de Engenharia de Software, SBC, pp. 134-149, 1990.
- [Leite00] Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N.: "A Scenario Construction Process", Requirements Engineering Journal, Vol.5, N° 1, pp. 38-61, 2000.
- [Leite05] Leite J.C.S.P., Doorn J.H., Kaplan G.N. , Hadad G.D.S., Ridaio M.N., "Defining System Context Using Scenarios", in Perspectives on Software Requirements, Kluwer Academic Publishers, pp.169-199, 2005.
- [Pressman06] Presuman, Roger, "Ingeniería de Software", Sexta edición, McGraw-Hill, México, 2006 ISBN 970-10-5473-3.
- [Parviainen05] Parviainen, P., Tihinen, M., van Solingen, R., "Requirements Engineering: dealing with the Complexity of Sociotechnical Systems Development", in Requirements Engineering for Sociotechnical Systems, J.L. Maté & A. Silva (eds.) , Chapter I, pp1-20, 2005, ISBN 1-59140-506-8.
- [Ruiz06] Ruíz, F, Hilera, J. R., "Using Ontologies in Software Engineering and Technology", in Ontology for Software Engineering and Software Technology, Calero, C., Ruíz, F., Piattini, M. (eds), Chapter II, Springer, pp 49-95, 2006, ISBN 3-540-34517-5.
- [Seidewitz03] Seidewitz, E., "What models mean". IEEE Software, 20:26-32, September 2003.
- [Sommerville02] Sommerville, Ian, "Ingeniería de Software", Sexta edición, Pearson Educación, México, 2002, ISBN 970-26-0206-8.
- [Young04] Young Ralph, "The Requirements Engineering Handbook", Artech House, 2004, ISBN 1-58053-266-7.