

## MEJORANDO EL NIVEL SEMÁNTICO DEL LÉXICO EXTENDIDO DEL LENGUAJE

Marcela Ridao <sup>(1)</sup> Jorge H. Doorn <sup>(1) (2)</sup>

(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: {mridao, 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. El uso de glosarios se ha difundido en el ámbito de la Ingeniería de Requisitos en los últimos años, tanto en la práctica como en la investigación. Por otra parte, se trata de una práctica común en otras actividades del desarrollo de software. En este proyecto se pretende analizar un modelo específico de glosario: el Léxico Extendido del Lenguaje con el fin de detectar mayor información que la que puede obtenerse de su mera lectura. Se pretende encontrar agrupamientos de símbolos que representen áreas relevantes dentro de un Universo de Discurso.

### INTRODUCCIÓN

Entender y elicitar las necesidades de los clientes-usuarios es una difícil tarea a la que deben enfrentarse habitualmente los desarrolladores de software. Cuanto más complejo es el contexto del problema, más difícil se hace la elicitación de los requisitos de software. A menudo, los ingenieros de requisitos deben convertirse casi en expertos en el dominio del problema durante la adquisición de conocimiento del contexto de la aplicación.

Uno de los objetivos de la Ingeniería de Requisitos (IR) es sistematizar el proceso de definición de requisitos [1] [2] [3] [4]. Para ello, se proponen cuatro actividades principales: elicitación, análisis, validación y comunicación de las necesidades, expectativas y restricciones de los clientes. Existe una actividad posterior a la definición de los requisitos del software, que se encarga de la administración de los cambios en los requisitos y la aparición de nuevos requisitos durante y después del proceso de desarrollo de software.

Debido a que, tanto los ingenieros de requisitos como los clientes-usuarios deben participar y colaborar para la definición de los requisitos de software, se hace necesario ganar un compromiso entre ambas partes. Para ello, muchos autores recomiendan el uso de modelos basados en lenguaje natural, que facilita la comunicación entre todos los involucrados en esta etapa del proceso de desarrollo de software.

Sin embargo, las representaciones basadas en lenguaje natural tienen un problema inevitable: la ambigüedad [5] [6]. Una especificación de requisitos ambigua puede llevar a dos o más implementados a interpretaciones diferentes y, en consecuencia, a códigos diferentes. Este problema se ve exacerbado si, por alguna razón, el o los autores de la especificación no están disponibles cuando los requisitos son interpretados durante la implementación. Esto es bastante común dado que los ingenieros de requisitos responsables de una especificación suelen dedicarse a proyectos diferentes y puede suceder que no estén disponibles para ser consultados por los implementadores [7].

La IEEE, en su norma Std 830-1998 señala que una especificación de requisitos será no ambigua si, y solo si, cada requisito tiene una única interpretación. Como mínimo, esto requiere que cada característica del producto de software sea descripta usando un único término. En los casos donde un término usado en un contexto particular podría tener múltiples significados, el término

debería ser incluido en un glosario donde su significado sea claramente especificado [8].

La construcción de un vocabulario que capture la jerga usada por los expertos del dominio ha sido propuesta por distintos autores [9] [10]. De hecho, varias experiencias han mostrado que un glosario del vocabulario de los clientes-usuarios es, en si mismo, una fuente de información para elicitar información del Universo de Discurso (UdeD) [11] [12] [13] [14] [15].

En este trabajo, se analizará el modelo particular de glosario utilizado: Léxico Extendido del Lenguaje (LEL). En la próxima sección, se presenta el modelo del LEL describiendo sus elementos y propiedades. Luego, se incluye una sección donde se presentan las hipótesis de este proyecto, así como las ideas iniciales para abordar el tema propuesto, arribándose a algunas conclusiones preliminares.

## **LEL: LÉXICO EXTENDIDO DEL LENGUAJE**

El Léxico Extendido del Lenguaje es una representación de los símbolos del lenguaje del dominio del problema, que intenta capturar el vocabulario de una aplicación. Su objetivo principal es que el ingeniero de requisitos conozca el lenguaje que habla el usuario, entendiendo los términos que utiliza, sin preocuparse por entender el problema [16] [17].

El LEL involucra la denotación y la connotación de cada símbolo descubierto como una palabra o frase relevante al dominio de la aplicación. El propósito de la construcción del léxico no sólo es habilitar una buena comunicación y acuerdo entre los clientes/usuarios y el equipo de ingeniería sino también facilitar la construcción de escenarios y ayudar a su descripción, facilitando la validación.

Este léxico se construye utilizando lenguaje natural y está compuesto, en primera instancia, por símbolos que pueden ser objetos activos o Sujetos (realizan acciones), objetos pasivos u Objetos (las acciones se realizan sobre ellos), Verbos (acciones del sistema) y Estados significativos del sistema [18].

Cada símbolo en el léxico tiene uno o más nombres o frases que lo identifican y dos tipos de descripciones, la noción y el impacto. La noción describe la denotación de la palabra o frase. Indica quién, cuándo ocurre, qué procesos involucra, qué significado tiene el símbolo, etc. El impacto describe la connotación del símbolo, es decir, su repercusión en el sistema. Esta descripción puede contener acciones que se ejecutan sobre otros objetos o que se aplican al que se está describiendo, situaciones derivadas de la que se está definiendo, etc. Cada entrada puede contener una o más nociones y uno o más impactos.

En la descripción de los símbolos deben cumplirse simultáneamente dos reglas básicas:

***Principio de circularidad:*** en la descripción de la noción o impacto de los símbolos se debe maximizar el uso de otros símbolos del léxico. De esta manera, el conjunto de símbolos determina una red, que permite representar al LEL mediante un hipertexto que puede ser navegado para conocer todo el vocabulario del problema.

***Principio del vocabulario mínimo:*** se debe minimizar el uso de símbolos externos al lenguaje de la aplicación. De este modo, se acota el lenguaje al menor conjunto de símbolos posible. Si se utilizan símbolos externos, éstos deben pertenecer al vocabulario básico del lenguaje natural que se está utilizando.

A continuación se presenta el modelo utilizado para representar los símbolos del LEL:

<p><b>LEL:</b> Representación de los símbolos en el lenguaje del dominio de la aplicación  Sintaxis:  <math>\{\text{Símbolo}\}_1^N</math></p>
<p><b>Símbolo:</b> Entrada del léxico que tiene un significado especial en el dominio de la aplicación  Sintaxis:  <math>\{\text{Nombre}\}_1^N + \{\text{Noción}\}_1^N + \{\text{Impacto}\}_1^N</math></p>
<p><b>Nombre:</b> Identificación del símbolo. Más de uno representa sinónimos  Sintaxis:  Palabra   Frase</p>
<p><b>Noción:</b> Denotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo  Sintaxis:  Sentencia</p>
<p><b>Impacto:</b> Connotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo  Sintaxis:  Sentencia</p>

Figura 1 – Modelo del Léxico Extendido del Lenguaje

donde:

**Sentencia** está compuesto por Símbolos y No-Símbolos pertenecientes al vocabulario mínimo,  
+ significa composición,  
{x} significa cero o más ocurrencias de x,  
() es usado para agrupamiento,  
| significa or, y  
[x] significa que x es opcional

## RESULTADOS ESPERADOS

Durante la realización de casos prácticos en los que se ha construido un LEL parece ocurrir que hay símbolos más vinculados a los restantes, ya sea por contener muchas referencias a otros símbolos, o por ser referenciados muchas veces por los demás.

Intuitivamente, se puede suponer que si el Universo de Discurso contiene distintas áreas de interés, fragmentos de la organización o subprocesos diferenciados, entonces se debería esperar cierto grado de acoplamiento mayor entre términos que describen sujetos, objetos, verbos o estados de un fragmento determinado que entre los términos correspondientes a áreas diferentes.

La teoría de grafos ha sido profusamente utilizada para visualizar, modelar y detectar este tipo de fenómenos en redes de diferente naturaleza [19].

Por lo anterior, este proyecto tiene como hipótesis central la suposición que a partir del análisis del grafo construido a partir del LEL, tomando como nodos los símbolos, y como arcos las referencias entre ellos, se podrá detectar alguna de las propiedades antes mencionadas, tales como símbolos muy relevantes, áreas de interés, subconjuntos destacados entre otras posibles.

Se plantea construir los grafos asociados a varios LEL correspondientes a diferentes casos y

estudiar los mismos utilizando las diferentes herramientas provistas por la teoría de grafos.

Obviamente, en caso de detectarse singularidades o estructuras destacadas en el grafo, se retornará al UdeD correspondiente para calibrar el grado de predictibilidad que estas estructuras tienen sobre aspectos semánticos ocultos total o parcialmente en las fuentes de información.

Se aspira lograr una definición de **cluster** dentro del grafo que representa los símbolos del LEL y las relaciones entre ellos, que permita detectar grupos de símbolos que representen áreas destacadas dentro del UdeD. Por un lado, se analizarán clusters basados en las distancias entre los nodos, y por otro se considerará la conectividad entre ellos.

En este estado del proyecto, se entiende por **distancia** entre dos nodos A y B el número de arcos que se recorren para llegar del nodo A al nodo B. Por ejemplo, en la figura 2, se presentan 4 nodos representando símbolos correspondientes al caso de estudio “Plan de Ahorro y Préstamo para la adquisición de automóviles 0 km.” [20]. El nodo correspondiente al símbolo Adherente tiene una distancia 3 al nodo correspondiente al símbolo Adjudicatario.

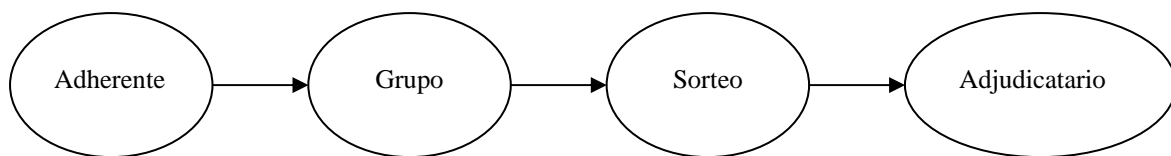


Figura 2 – Distancia entre nodos

Del mismo modo, se llamará **conectividad** al número de caminos independientes entre dos nodos.

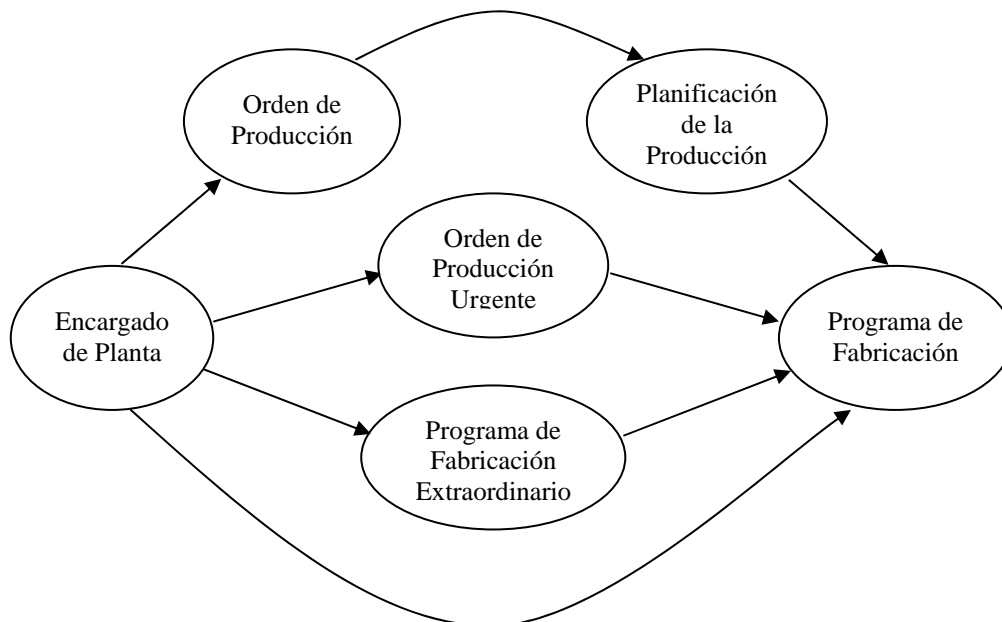


Figura 3 – Conectividad entre nodos

En la figura 3 se presenta un ejemplo donde los nodos representan símbolos correspondientes al caso de estudio “Producción de Cajas de Cartón Corrugado” [21]. En este ejemplo, la conectividad entre los nodos Encargado de Planta y Programa de Fabricación es 4, ya que existen cuatro caminos independientes desde un nodo al otro.

Se presume que, si se encuentran diferentes clusters que agrupen nodos ya sea por distancia, o por conectividad, cada uno de estos clusters podría estar señalando un área de interés dentro del

Universo de Discurso bajo estudio.

También se espera encontrar una semántica dentro del UdeD cuando dos clusters de conectividad diferente se agrupen de tal manera que el de mayor conectividad sea un subconjunto propio del de menor conectividad. Esto ya ha sido observado en casos reales.

## CONCLUSIONES

Los estudios preliminares ya realizados resultan promisorios, ya que se han encontrado regularidades en las estructuras de conectividad de los símbolos de los léxicos extendidos del lenguaje.

Se planifica utilizar los algoritmos conocidos para el caso de distancias y desarrollar algoritmos específicos para el cálculo de conectividades y otras eventuales nociones de agrupamiento.

## REFERENCIAS

- [1] Maculay, L. (1993). Requirements Capture as a Cooperative Activity. Proceedings of IEEE International Symposium on Requirement Engineering. IEEE Computer Society Press, San Diego, Ca. 174-181.
- [2] Reubenstein, H. B., Waters, R.C. (1991). The requirements apprentice: Automated assistance for requirements acquisition. IEEE Transaction on Software Engineering, 17 (3), 226-240.
- [3] Jacobson, Y., Christerson, M., Jonsson, P., Overgaard, G. (1992). Object-Oriented Software Engineering – A Use Case Driven Approach. Addison Wesley, Reading, MA, ACM Press, New York.
- [4] Bubenko, J.A., Wrangler, B. (1993). Objective driven capture of business rules and information systems requirements. Proceedings de IEEE Conference on Systems, Man and Cybernetics.
- [5] Jackson, M. (1995). Software Requirements & Specifications. A lexicon of practice, principles and prejudices. Addison Wesley, ACM Press.
- [6] Sommerville, I. (2001), Software Engineering. Reading, MA: Addison Wesley, 6th Edition.
- [7] Berry, D.M., Kamsties, E. (2004) Ambiguity in Requirements Specification. In Leite, J.S.C.P., Doorn, J.H (Eds.) Perspectives on Software Requirements. Kluwer Academic Press. 7-44.
- [8] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specification. The Institute of Electrical and Electronics Engineers, Inc. ISBN 0-7381-0332-2.
- [9] Arango, G., Schafer, W., Prieto, R. (1993) Domain Analysis Methods – Software Reusability. Ellis Horwood Ltd.
- [10] Leite, J.C.S.P., Franco, A.P.M. (1990). O Uso de Hipertexto na Elicitação de Linguagens da Aplicação, Anais de IV Simpósio Brasileiro de Engenharia de Software, SBC, Brazil. 134-149.
- [11] Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C. (1999). Guiding Use Case Authoring: Results of an Empirical Study. Proceedings International Symposium On Requirements Engineering. IEEE Computer Society Press, Limerick-Ireland. 36-43.
- [12] Rolland, C., Ben Achour, C. (1998). Guiding the construction of textual use case specifications. Data & Knowledge Engineering, 25, 125-160.
- [13] Oberg, R., Probasco, L, Ericsson, M. (1998). Applying Requirements Management with Use Cases. Rational Software Corporation.
- [14] Regnell, B. (1999). Requirements Engineering with Use Cases – a Basis for Software Development, Doctoral Thesis, Department of Communication Systems, Lund University.
- [15] Prakash, S., Aurum, A., Kox, K. (2004). Requirements Engineering Practice in Pharmaceutical and Healthcare Manufacturing. Proceedings of the 11th Asia-Pacific Software Engineering Conference. 402-409.
- [16] Leite, J.C.S.P., Franco, A.P.M. (1993) A Strategy for Conceptual Model Acquisition, IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press. 243-246.
- [17] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A. (1997). Enhancing a Requirements Baseline with Scenarios, Requirements Engineering Journal, 2(4), , pp. 184-198.
- [18] Leite, J.C.S.P., Doorn, J., Kaplan, K., Hadad, G., Ridao, M. (2004) Defining System Context Using Scenarios. In Leite, J.S.C.P., Doorn, J.H (Eds.) Perspectives on Software Requirements. Kluwer Academic Press. 169-199.
- [19] Basu, A., Blanning, R. W. (2007) Metagraphs and Their Applications. Springer Science+Business Media, LLC.
- [20] Mauco, V., Ridao M., del Fresno, M., Rivero, L., Doorn, J.H. (1997). Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro. Reporte técnico, ISISTAN, UNCPBA, Tandil, Argentina.
- [21] Doorn, J., Ridao, M., Leonardi, C. (2002). Ingeniería de Requisitos, Caso de estudio: Cajas. Trabajo de Cátedra, UNCPBA, Tandil, Argentina