

Transformación de lenguaje natural a controlado en la educación de requisitos: una síntesis conceptual basada en esquemas preconceptuales

Transforming natural language into controlled language in requirements elicitation: a pre-conceptual-schema-based conceptual synthesis

Bell Manrique Losada^{1}, Carlos Mario Zapata Jaramillo²*

¹Facultad de Minas. Universidad Nacional de Colombia. Bloque M8A. Carrera 80 No. 65-223. Medellín, Colombia.

²Facultad de Ingeniería. Universidad de Medellín. Calle 87 N.° 30-65 Bloque 4 Oficina 113. Medellín, Colombia.

(Recibido el 10 de enero de 2012. Aceptado el 09 de octubre de 2013)

Resumen

La educación de requisitos implica la identificación, captura e integración de requisitos, bajo la intervención de un grupo de analistas e interesados, generando descripciones textuales o gráficas que plasman los conceptos más relevantes del dominio del interesado, para el desarrollo de una aplicación de software. Tradicionalmente, las etapas de identificación y obtención de requisitos (que se expresan en lenguaje natural) se ejecutan a partir de técnicas que requieren alta intervención del analista y amplio conocimiento del contexto/dominio del problema. Lo anterior implica un proceso subjetivo, ambiguo, con errores, que ocasiona pérdidas en la generación de modelos iniciales del dominio (especificados en un lenguaje controlado). En este artículo se realiza una síntesis de los acercamientos conceptuales y tendencias encontrados en la literatura referentes a la transformación de lenguaje natural a lenguaje controlado en la educación de requisitos. Finalmente, se propone un esquema pre-conceptual que representa el marco conceptual del proceso de transformación.

----- *Palabras clave:* Educación de requisitos, lenguaje natural, lenguaje controlado

Abstract

During the requirements elicitation process, a group of analysts and stakeholders identify, capture, and integrate requirements. Textual or

* Autor de correspondencia: teléfono: + 57 + 4 + 340 55 28, fax:+ 57 + 4 + 340 52 16, correo electrónico: bmanrique@udem.edu.co (B. Manrique)

graphic descriptions capturing the most relevant concepts from the domain of a software application development are generated. Commonly, the initial phases—identifying and capturing requirements expressed in natural language—are executed by using techniques in which high analyst intervention and comprehensive knowledge of the context and the problem domain are required. Thus, a subjective, ambiguous, and error-prone process is implied, causing losses in the generation of the initial domain models (specified in a controlled language). In this paper we provide a synthesis of trends and conceptual approaches found in the state of the art concerning the natural language transformation into controlled language during the requirements elicitation process. Finally, we propose a pre-conceptual schema for representing the conceptual framework of the transformation process.

----- *Keywords:* Requirements elicitation, natural language, controlled language

Introducción

El proceso mediante el cual un grupo de analistas e interesados realiza las tareas de identificación, captura e integración de requisitos se conoce como educación de requisitos. Como resultado de este proceso, en el marco del desarrollo de una aplicación de software, se generan descripciones de los conceptos más relevantes del dominio del interesado, las cuales pueden ser textuales, gráficas o una combinación de las dos. Así, se puede afirmar que educir implica descubrir las necesidades y expectativas de los interesados en cuanto a un problema a sistematizar, las cuales se expresan en lenguaje natural. Se pretende su traducción a una especificación de software durante la fase de análisis posterior.

La educación implica la obtención o transferencia de conocimiento, dependiendo de cuál sea la fuente del requisito: de forma interactiva, entre analistas e interesados, cuando es posible aplicar técnicas de captura directas como entrevistas o cuestionarios; de forma unidireccional, cuando se obtienen a partir de técnicas de revisión u observación, como el análisis de documentación técnica organizacional. Las etapas de identificación y obtención de requisitos, tradicionalmente, se ejecutan a partir de técnicas que demandan, además de una alta intervención del analista, un amplio conocimiento del contexto y del dominio

del problema. El anterior escenario implica un proceso subjetivo, ambiguo, con errores, que genera pérdidas de tiempo y dinero en la generación de modelos iniciales del dominio, los cuales se especifican en un lenguaje controlado. Como consecuencia, los productos intermedios y finales del desarrollo se asocian con altos grados de insatisfacción de los interesados. Como un aporte en la mejora de estas debilidades, diversos investigadores realizan propuestas enfocadas hacia: el análisis de información descrita en lenguaje natural obtenida tradicionalmente por medio de entrevistas y reuniones con el interesado; nuevas técnicas de apoyo en la educación que permiten mayor intervención del usuario en la validación de requisitos; esquemas intermedios de representación de necesidades y expectativas del usuario en un lenguaje cercano al natural; propuesta de lenguajes naturales controlados que permiten procesar documentos de requisitos expresados en lenguaje natural; entre otros.

En este artículo se propone una síntesis de los acercamientos conceptuales que se encuentran en la literatura referentes a la transformación de lenguaje natural a lenguaje controlado en la educación de requisitos, tomando como base una metodología para la revisión de la literatura [1]. A partir de la revisión anterior, se propone un esquema preconceptual que representa el conocimiento del dominio de este proceso

de transformación, por medio de conceptos y relaciones entre conceptos.

En el siguiente apartado se presenta la fundamentación en términos teóricos y conceptuales, se continúa con una descripción de las diferentes propuestas sistematizadas por medio de la revisión de literatura, posteriormente se propone el esquema de síntesis de los principales conceptos que especifican el proceso de transformación de lenguaje a lenguaje controlado para la educación de requisitos y se finaliza con las conclusiones y el trabajo futuro.

Marco teórico

Educción de Requisitos

La educación de requisitos es una de las primeras fases de la *Ingeniería de Requisitos* para el desarrollo de software, cuyo objetivo principal es descubrir todos los requisitos que la futura aplicación de software debe satisfacer para que se considere de calidad. Para lograrlo, se deben llevar a cabo, de manera iterativa e incremental, dos actividades: *educación de requisitos* y *análisis de requisitos*, involucrando el lenguaje natural y un lenguaje de modelado, respectivamente [2].

La fase de educación de requisitos tiene diferentes actividades, que incluyen: entendimiento del dominio de aplicación, captura y clasificación de requisitos, establecimiento de prioridades, resolución de conflictos y negociación de los requisitos del sistema [3]. La educación se relaciona con la acción de comunicación analista-interesado, que busca recuperar la información esencial y relevante acerca del dominio, que se convertirá en la base de los requisitos y la cual se extrae de los interesados (cliente, usuario final, experto del dominio, etc.).

Lenguaje Natural y Lenguaje Controlado

La *lengua* y el *habla* son las dos entidades complementarias para expresar un lenguaje [4]. Estas dos entidades conforman el sistema de signos de que dispone una sociedad (y no un

individuo aislado) para comunicarse. Esos signos conforman oraciones gramaticalmente bien formadas (*i.e.*, oraciones que cumplen reglas fonéticas, léxicas, sintácticas y semánticas) [5]. En la ingeniería de requisitos, donde son determinantes la exactitud y la precisión en los discursos en lenguaje natural, se deben establecer reglas que determinen las relaciones entre ciertas expresiones bien formadas y los sentidos que ellas pretenden transmitir [6].

Por lenguaje natural (LN) se entiende la lengua que utiliza normalmente una comunidad para su comunicación interna [7]. El LN se caracteriza por su enorme capacidad y riqueza comunicativa, su flexibilidad y la posibilidad de jugar con las palabras y con las expresiones, produciendo metáforas y ambigüedades. De lo anterior se deduce que, si bien es un instrumento idóneo para ciertos propósitos, no lo es igualmente para áreas científicas o ingenieriles donde se requiere alta exactitud y precisión [2].

La gran mayoría de requisitos se escribe en LN [6]. Para el analista es muy importante identificar los conceptos y relaciones entre conceptos que emplea el interesado. Estos conceptos y relaciones construyen la base del lenguaje común que debe entender el analista y el experto. El LN es altamente informal por naturaleza [2], lo que implica mayor intervención y transformación para las tareas posteriores de análisis y diseño de una aplicación de software.

Un Lenguaje Controlado (LC), es un subconjunto del LN con sintaxis, semántica y/o terminología restringidas [8]. Algunos autores lo definen a partir de un conjunto de reglas que debe cumplir el lenguaje, así como el glosario que se debe utilizar [9].

Lingüística Computacional

El *lenguaje* se estudia desde diferentes disciplinas orientadas hacia la *lingüística*. La lingüística estudia todos los hechos y fenómenos relacionados con el LN. El objetivo de la *lingüística* es producir modelos que aproximen el comportamiento humano en sus tareas básicas:

leer, escribir, escuchar y hablar. La lingüística se enfoca en el estudio de los signos lingüísticos e incluye semántica, sintaxis y pragmática [10]. Una de las disciplinas que estudia el lenguaje es la *lingüística computacional*, cuyo propósito es desarrollar una teoría computacional del lenguaje, a partir de las nociones de algoritmos y estructuras de datos de las ciencias de la computación [11].

El término *lingüística computacional* (en inglés *computational linguistics*), se refiere al campo interdisciplinario entre la lingüística, fonética, ciencias de la computación, ciencias cognitivas, inteligencia artificial y lógica formal [12] que se concentra en el estudio del LN, tal como lo hace la lingüística tradicional, pero usando equipos de cómputo como herramienta para modelar fragmentos de teorías lingüísticas con un interés particular [13].

Metodología de Kitchenham

Existe una metodología que integra un conjunto de directrices para la realización de revisiones sistemáticas de la literatura en ingeniería de software [1]. Esta metodología enfatiza que, inicialmente, se debe confirmar la necesidad de realizar la revisión, resolviendo una pregunta de investigación. De esta forma, se enfoca la búsqueda siguiendo un plan de revisión. En él se definen los procedimientos de revisión básicos, entre los que se destaca el protocolo de revisión, apoyado en un método con el cual se estructura la búsqueda, haciendo que en los procesos de extracción de datos, análisis y síntesis se identifiquen. Finalmente se obtienen las propuestas que respondan a las preguntas planeadas.

Transformación de LN a LC: problemas referentes

En el marco de la ingeniería de requisitos, para iniciar el proceso de educación, se requiere descubrir y obtener el máximo de información sobre un contexto a partir de un discurso. Una vez se consolida ese discurso, el analista

representa, mediante un modelo, el ámbito del dominio y su solución, empleando para ello esquemas conceptuales. Para desarrollar un esquema conceptual, el analista o diseñador debe identificar ciertos elementos conceptuales, identificar las relaciones entre ellos y entenderlas, para luego elaborar la representación en un lenguaje de modelado [14]. Este proceso se puede visualizar como una ‘traducción’, de un lenguaje base a otro diferente, tal que el traductor debe reconocer y entender símbolos expresados en un LN de un universo de discurso, para elaborar un conjunto de símbolos definidos en un lexicón de un lenguaje de modelado [10].

En este proceso de traducción, el analista representa en modelos técnicos las necesidades y expectativas del interesado. Los interesados, por su parte, validan los requisitos plasmados en dichos modelos, sin comprenderlos suficientemente, ya que se encuentran en un lenguaje técnico que se aleja del natural. Esta comunicación entre los participantes (interesado-analista), se torna más débil aún por la diferencia de formación y experiencia entre ellos [15], lo que genera problemas en cuanto a su correcta validación, el alcance de los modelos generados y, finalmente, los altos costos de ejecución de la educación. Por otro lado, tradicionalmente, la obtención de requisitos parte de la aplicación de técnicas de captura como entrevistas y diseño de aplicaciones conjuntas [16], u otras técnicas enfocadas hacia el análisis de escenarios [17]. A partir estas técnicas, en este proceso se pierde tiempo, secuencia y concisión, dado que los interesados tienden a dilatar sus intervenciones y la entrega de información, generando mayores costos y tiempos. El compendio de información obtenida y las descripciones del dominio de aplicación, tienen los problemas propios del LN: mucha información, uso indiscriminado de sinónimos y ambigüedades, entre otros. No es común educir requisitos a partir de fuentes como la documentación técnica de la organización, la cual incluye información como manuales de procedimientos, reglamentos y estatutos. Esta educación permitiría, principalmente: la comprensión y descripción detallada de la propia

organización y el papel que representa el sistema en ese contexto [18], la comprensión del dominio del interesado y la generación de modelos iniciales del dominio del problema.

A pesar de los acercamientos propuestos para reducir la brecha entre los universos de discurso del interesado y del analista, todavía se exige alta intervención de este último en el proceso de educación, en la conversión entre la descripción de requisitos y modelos de diseño y en la guía del proceso. Es necesario lograr acercamientos entre los lenguajes controlados que existen para especificar los requisitos y el LN del interesado, el cual se puede traducir directamente a partir de documentación técnica y, así, conducir a las etapas posteriores del proceso de desarrollo, que se realiza actualmente de forma automática [19].

Proceso de revisión sistemática y aproximaciones actuales

La revisión sistemática de literatura se realizó siguiendo la adaptación a una metodología existente [1], la cual se plasma en un protocolo de revisión que incluye las fuentes de información, las expresiones de búsqueda, los criterios de inclusión/exclusión y la eliminación de duplicados.

Se definieron las *fuentes de información* (Bases de datos: SCOPUS, ScienceDirect, EBSCO, IEEE; Bibliotecas Digitales: ACM, BUCM, CitiSeer, INFOMINE; Catálogos: Latindex, Rebiun y Biblioteca Virtual; y los Directorios: CCSB, DOAJ, DoCIS, OCLC, REDALYC, Worldcat). Se seleccionaron de forma manual los *términos y expresiones de búsqueda*, a partir de un grupo de términos candidatos que respondían a la pregunta de investigación. Estos términos conforman las expresiones de búsqueda o palabras clave, que son determinantes para optimizar las búsquedas y determinar los estudios relevantes. Estas expresiones se resumen en: Transformación LN; LN a LC; Transformación LN a LC; Conversión LN a LC; Transformación de LN a LC en

educación de requisitos; Procesamiento LN en educación de requisitos.

Los *criterios de búsqueda e inclusión/exclusión*, se definieron recopilando los artículos encontrados en la fase de búsqueda: posteriormente, se aplicaron tres criterios de filtro en el título y el resumen de cada una: ser un estudio del área de ingeniería de software; estar publicado en inglés o español y enfocarse en la educación de requisitos. Finalmente se aplicaron los siguientes criterios de inclusión y exclusión para definir la relevancia del estudio, a partir de la revisión de su resumen y conclusiones: explica el contexto en el que se realizó; describe los objetivos que lo motivaron; propone algún marco de trabajo/herramienta/modelo/método para la transformación de LN a LC en la educación de requisitos.

Como resultado de la primera etapa de búsqueda, se obtuvieron los resultados de la tabla 1, clasificados por fuente de información: Del total de 381.833, 15.920 son en español. Teniendo en cuenta que esta iteración refleja que las búsquedas ejecutadas con las expresiones en inglés corresponden al 96% del total, la siguiente iteración de la búsqueda sólo se hace con las expresiones e inglés.

Tabla 1 Resultados de búsqueda por fuente de información en etapa 2

<i>Fuente de búsqueda</i>	<i>Cantidad de publicaciones</i>
ACM	283.578
SCOPUS	75.435
The Collection of Computer Science Bibliographies	13.908
ScienceDirect	5.916
CitiSeer	1.857
Rebiun: Red de bibliotecas universitarias españolas	955
Biblioteca de la Universidad Complutense de Madrid	184
Total General	381.833

La siguiente iteración generó los resultados de la tabla 2.

Tabla 2 Resultados de búsqueda por fuente de información en etapa 2

Fuente de búsqueda	Cantidad de publicaciones
Google Scholar	150.917
Worldcat	18.389
Microsoft Academic Search	5.993
The IEEE Transactions on Software Engineering (TSE)	5.366
Base	4.093
Scirus	3.289
EBSCOhost	1.011
Otras Fuentes	1.309
Total General	190.367

En la aplicación del protocolo de revisión, se tomaron los primeros 300 resultados de cada fuente, se eliminaron duplicados y se obtuvieron 10.504. Luego de aplicar los dos primeros criterios de filtro a los resultados anteriores, se obtuvieron 193 estudios. De los anteriores, se preseleccionan 81 que corresponden a los que cumplan con el tercer criterio. Finalmente, luego de aplicar los criterios de inclusión y exclusión, se obtuvieron 28 estudios, los cuales se consideran estudios

primarios. La síntesis de los estudios primarios se realizó basada en unas categorías de clasificación que se establecieron para identificar la entrada o fuente del método usado en cada propuesta y el tipo de resultado o salida del proceso. Asimismo, para cada categoría de clasificación, se especifican los siguientes aspectos: tipo de aporte lingüístico del trabajo (en términos gramaticales, sintácticos, semánticos y/o pragmáticos), idioma de los documentos fuente para el procesamiento; enfoque y descripción de la propuesta; y herramienta o método propuesto.

De esta manera, en la tabla 3 se muestra la síntesis de los estudios en la categoría *i* enfocados hacia el procesamiento de LN desde requisitos educidos (los que comúnmente se encuentran en documentos de requisitos o *Software Requirements Specification* SRS); en la tabla 4 se presenta la síntesis de los estudios clasificados en la categoría *ii*, orientados hacia el procesamiento de LN-controlado (cuya entrada son textos fuente en lenguaje con algún tipo de intervención); y, en la tabla 5, se presenta la síntesis de los trabajos de la categoría *iii*, orientados hacia el procesamiento de LN desde requisitos textuales (directamente desde discurso escrito, sin ninguna intervención).

Tabla 3 Síntesis de Estudios Primarios en categoría i

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[20], [21], [22]	Inglés	No especificado	Desarrollo de software a la medida y desarrollo orientado al mercado	Técnica para análisis de similaridad. Aporte en área de 'Ingeniería Lingüística'.	ReqSIMILE Tool
[23]	Inglés	Gramática Restringida	<u>Análisis</u> de documentos de requisitos	Gramática para el dominio de automovilismo	No especificado
[24]	Inglés	Gramática de dos niveles	Especificación <u>formal</u> ejecutable	Gramática y estructura en XML. Aporte en área de 'PLN Contextual'.	VDM Toolkit
[25]	Inglés	No especificado	<u>Análisis</u> y evaluación de calidad de los SRS en LN.	Educación asistida por herramientas. Máquinas de estados para verificación de requisitos.	Requirements Analysis Tool (RAT)
[26]	Inglés	No especificado	Generación de modelos OO	Sistema de PLN para generación de requisitos OO en LN.	NL-OOPS LOLITA

... continuación Tabla 3

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[27]	Inglés	Tokenización y análisis morfo-sintáctico	Modelado de requisitos, validación formal	Generación automática de modelos.	CIRCE
[28]	Inglés	No especificado	Detección ambigüedades	Técnicas de aprendizaje de máquina para detectar ambigüedad en frases	Nocuous Ambiguity Identification (NAI)
[29]	Inglés	No especificado	Ingeniería de requisitos orientada a aspectos	Método de identificación temprana de aspectos, con técnicas de minería.	EA-Miner
[30]	Inglés	No especificado	Extracción de objetos desde SRS y generación modelo OO.	Analizador de SRS en LN para generar modelos OO	EDGE
[31], [32]	Inglés	Formulación semántica de reglas de negocio	Representación formal de LN a reglas de negocio	Estructura semántica de declaraciones y definiciones, a partir de lógica formal	NL2SBVR tool RuleGenerator
[33]	Inglés	No especificado	Formalización y validación de requisitos	Categorización automática de requisitos y formalización en UML	EuRailCheck
[34]	Inglés	Extractor de características de la oración	Evaluación automática de calidad de SRS	Modelo de calidad para requisitos textuales y sistema de clasificación automático	No especificado

Tabla 4 Síntesis de Estudios Primarios en categoría ii

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[35]. [36]	Español	Lenguaje Controlado: UN-Lencep	Educción semi-automática	Modelo intermedio (Esquema preconceptual) y su transformación a modelos UML	UNC-Diagrammer
[37]	Inglés	Patrones lingüísticos	Modelo para especificar requisitos.	Identificación de patrones lingüísticos comunes desde SRS.	ProjectIT-Requirements
[38]	Inglés	Gramática de superficie - DCG (definite-clause grammars)	Validación y especificación de requisitos	Técnica de agregación (análisis e integración de redundancia) de LN, para la validación y especificación	VINST-system
[39]	Inglés	Lenguaje Controlado: ACE estructuras de representación del discurso	Verificación de objetivos	Herramienta para la descripción de requisitos funcionales y su traducción a PROLOG.	Lexical Editor and Spelling Checker Attempto (ACE)
[40]	Inglés	Lenguaje Controlado Gramática especial en inglés	Especificación formal de requisitos	Uso de modelos formales algebraicos para ejecución automática de Casos de Uso	Generador CNL-grammar CNL knowledge-bases

... continuación Tabla 4

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[41]	Inglés	No especificado	Análisis de consistencia	Chequeo de consistencia de requisitos, aplicando un método formal	No Especificado
[42]	Inglés	No especificado	Generación LN	Construcción de sistemas de generación de LN	No Especificado
[43]	Inglés	No especificado	Verificación formal	Integración de ingeniería de requisitos y dirigida por modelos. Verificación de modelos	Wiki-based CSCW (Computer Supported Cooperative Work) Tool
[44]	Inglés	Lenguaje declarativo de bajo nivel Gramática estática/ dinámica	Generación de código ejecutable	Especificación de escenarios desde una GUI, y conversión a diagramas de secuencia	Play-Engine GUIEdit
[45]	Inglés	No especificado	Identificación inicial de clases-objetos-relaciones	Identificación automática de clases a partir de requisitos plasmados en descripciones de casos de uso	Class-Gen
[46]	Inglés	Lenguaje (ontología), para representar requisitos	Definición de un patrón para representar requisitos	Lenguaje de especificación de requisitos y meta-modelo.	ProjectIT-RSL Tool
[47]	Inglés	No especificado	Especificación formal desde procesos de negocio	Descripciones semánticas de modelos de procesos de negocio en BPMN	ProcessSEER
[48]	Inglés	Gramática ANLT	Especificación formal de requisitos, calidad de SRS	Sistema que detecta ambigüedades en SRS y genera expresiones en lógica formal.	No Especificado
[49]	Inglés	Esquema de traducción LNC-Lógica Formal	Tratamiento de inconsistencias en especificación de requisitos	Técnicas de análisis y traducción de LNC a lógica formal para identificar inconsistencias en SRS.	Framework CARL
[50]	Inglés	LN con Gramática controlada	Identificar inconsistencias y ambigüedades en SRS en LN	Método semi-automático que analiza SRS desde gramática de restricciones y genera modelo de análisis OO	Dowser
[51]	Inglés	Parametrización semántica	Especificación de objetivos sin ambigüedad	Especificación de definición del dominio desde objetivos, en descripciones lógicas.	Cerno framework

... continuación Tabla 4

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[52]	Inglés	Técnica de Posicionamiento de conceptos y <i>Key-Word-In-Context</i>	Traducción semi-automática, identificación de inconsistencias en requisitos	Técnica para asistir la generación de modelos UML desde LN normalizado	UMGAR
[53]	Inglés	Lenguaje natural estructurado	Traducción de especificaciones en LN a lenguaje formal	Técnica de traducción de descripciones en LN estructurado a especificación formal	SNL2Z
[54]	Inglés	Lenguaje Natural Controlado	Especificación y validación de requisitos	Herramienta de especificación y validación de requisitos, a partir de un LN-controlado	ProjectIT-Studio/ Requirements plugin
[55]	Inglés – Coreano	No especificado	Análisis de requisitos	Método de clasificación automático de requisitos, en categorías de análisis.	Internet-based requirements analysis-supporting system

Tabla 5 Síntesis de Estudios Primarios en categoría iii

Estudio Primario	Idioma	Aporte lingüístico	Enfoque Propuesta	Descripción de la propuesta	Herramienta
[56]	Inglés	Estructura de diálogos	Mejorar el proceso de educación en relación con la calidad en la orientación a los usuarios	Técnica de asignación de restricciones al lenguaje natural, desde la teoría de diálogo.	No especificado
[57]	Inglés	Gramática <i>McCord's Slot Grammar</i>	Estructura semántica y sintáctica de textos, adquisición de conocimiento	Educación de conocimiento desde frases textuales estructuradas	Herramienta semi-automática NIAM
[58]	Alemán	Teoría lingüística Modelo de gramática NTMS	<i>Tagging</i> e interpretación de textos en LN	Análisis y traducción de especificaciones textuales en un esquema de pre-diseño conceptual	NIBA-TAG

Esquema pre-conceptual como síntesis conceptual

Un esquema pre-conceptual (EP) constituye una forma gráfica de representar un discurso [59] y resume las principales características del dominio de un problema particular. Un discurso expresado mediante EP se suele desarrollar de manera conjunta entre analistas e interesados, facilitando así el entendimiento y conocimiento común del contexto.

Un EP es un modelo gráfico que presenta cierta unión de las características estructurales y dinámicas de un contexto, por medio de un conjunto de símbolos que representan: los conceptos (sustantivos o sintagmas nominales); relaciones estructurales (verbos que generan conexiones permanentes entre conceptos); relaciones dinámicas (verbos que denotan acciones u operaciones en el discurso); implicaciones (relaciones de causa-efecto); condicionales (expresiones entre conceptos y operadores que

actúan como precondiciones); conexiones (flechas de unión entre conceptos y relaciones dinámicas y estructurales); y referencias (permiten ligar elementos físicamente distantes en el diagrama).

El proceso de revisión sistemática de literatura realizado permitió reconocer conceptualmente el dominio de la transformación de lenguaje natural a lenguaje controlado, con propósitos de educación de requisitos. En este sentido, la revisión permitió delimitar en términos de

conceptos, relaciones dinámicas y relaciones estructurales, cómo ocurre dicha transformación y el alcance que tiene, reflejado en propuestas, métodos, herramientas y orientación lingüística.

La figura 1 muestra el EP propuesto, que sintetiza los conceptos encontrados en la revisión de literatura para representar el conocimiento asociado con la transformación de lenguaje natural a un lenguaje controlado en la educación de requisitos.

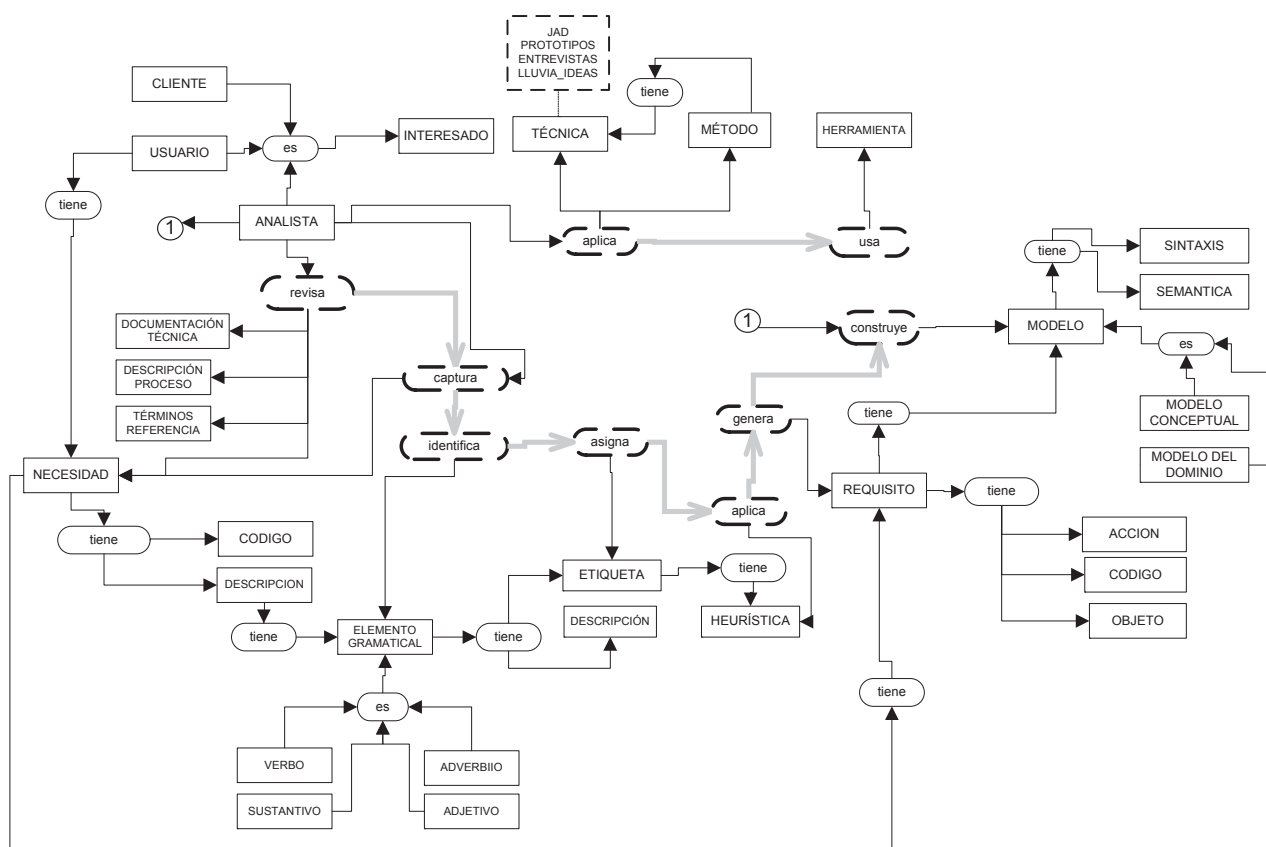


Figura 1 Esquema pre-conceptual propuesto (elaboración propia)

El EP se puede leer de la siguiente manera: en la Educación de Requisitos un INTERESADO puede ser el CLIENTE, el USUARIO o el ANALISTA. Un USUARIO tiene una NECESIDAD. El ANALISTA revisa DOCUMENTACIÓN TÉCNICA, DESCRIPCIONES DE PROCESOS o TÉRMINOS DE REFERENCIA y captura

una NECESIDAD a partir de esta revisión. La NECESIDAD tiene un código y una descripción. Luego de capturarlas, el ANALISTA identifica, de la descripción de la necesidad, sus ELEMENTOS GRAMATICALES. Los elementos gramaticales pueden ser: VERBO, SUSTANTIVO, ADVERBIO o ADJETIVO. El

elemento gramatical, a su vez, tiene una *etiqueta* y una *descripción*. Posteriormente, el ANALISTA asigna una ETIQUETA a cada elemento gramatical identificado y aplica una HEURÍSTICA particular según sea la ETIQUETA. Inmediatamente después, genera un REQUISITO, el cual tiene una *acción*, un *código* y un *objeto* asociado. El ANALISTA, finalmente, construye un MODELO que tiene una *sintaxis* y una *semántica* particular.

De esta manera, la NECESIDAD inicial que identifica el ANALISTA, tiene un conjunto de REQUISITOS asociados, los cuales se plasmarán en un MODELO, que puede ser CONCEPTUAL o de DOMINIO. El ANALISTA, para todo el proceso, también aplica un MÉTODO, el cual tiene un conjunto de TÉCNICAS, apoyadas en el uso de HERRAMIENTAS.

El EP presentado describe el dominio en cuestión, según la orientación de este tipo de esquemas conceptuales. Dicho esquema está dispuesto como modelo del dominio y ofrece una representación del conocimiento. Este tipo de síntesis sirve para entender en términos conceptuales cómo ocurre el proceso, cuáles son sus causas y consecuencias, qué tipo de resultados generan las actividades que se realizan y cuál es la tendencia de los aportes de los investigadores.

Conclusiones y trabajo futuro

Empleando un protocolo de revisión sistemática se pudo elaborar una síntesis conceptual de los estudios primarios que tratan sobre la transformación de lenguaje natural a lenguaje controlado en la educación de requisitos. Esta síntesis se basa en las siguientes categorías de clasificación: i) Cuando el procesamiento se realiza sobre requisitos educidos, es decir, aquellos que comúnmente se encuentran en documentos de requisitos o *Software Requirements Specification* (SRS); ii) cuando el procesamiento se realiza a partir del uso de un lenguaje natural-controlado, el cual es una modificación estructural del lenguaje natural; y, iii) cuando el procesamiento se realiza sobre el lenguaje natural de requisitos textuales (directamente desde discurso escrito).

En el análisis de los trabajos encontrados en cada categoría, se encontró que son considerables los aportes que se intenta hacer desde cada punto de vista y con orientaciones particulares. La mayoría de ellos buscan en los requisitos: mejorar su calidad, manejar los diferentes formalismos entre niveles de especificación, definir especificaciones formales y acercarse hacia su conversión automática en un lenguaje controlado o modelo formal, entre otros. La mayoría de los estudios proponen herramientas que apoyan ciertas actividades en el proceso de educación, generalmente orientadas hacia el análisis, validación y verificación de requisitos. Se puede identificar, en general, que los autores intentan definir y aplicar buenas prácticas en la educación, análisis y documentación de requisitos escritos en LN.

Dentro de los estudios categorizados se aborda el tema de la calidad de los requisitos, a partir de la aplicación de técnicas del procesamiento del LN, con orientaciones particulares hacia la clasificación del texto, la guía asistida en la especificación a partir de herramientas colaborativas, el entendimiento conceptual y aseguramiento de no-ambigüedades en los documentos de requisitos. Más del 90% de los aportes se logró con el procesamiento de LN en idioma inglés, aunque también hay acercamientos en Coreano, Alemán y Español.

Sigue existiendo una brecha entre el LN textual, en el que los *stakeholders* expresan sus necesidades y expectativas, y el lenguaje que se utiliza en la especificación de requisitos o documentos de requisitos, donde se requiere total intervención del analista en este proceso.

Finalmente se propuso, como aporte en la representación de conocimiento asociado a la transformación de LN a LC, un EP. Este EP sintetiza los conceptos encontrados en la revisión de literatura y su clasificación según las categorías definidas, resaltando dos aspectos importantes: primero, los aspectos dinámicos y estáticos de los requisitos, y segundo, la estructura funcional y lógica del proceso de transformación.

Agradecimientos

Este trabajo se enmarca dentro de los resultados obtenidos en el proyecto de investigación ‘Revisión de Literatura en Transformación de Lenguaje Natural a Lenguaje Controlado en la Educación de Requisitos’, cofinanciado entre la Universidad Nacional de Colombia, Sede Medellín y la Universidad de Medellín, Colombia.

Referencias

1. B. Kitchenham. “Procedures for Performing Systematic Reviews”. *Joint Technical Report*. Vol. 0401. 2004. pp. 7-21
2. K. Li, R. Dewar, R. Pooley. “Requirements capture in natural language problem statements”. *Technical Report Series No. 0023*, Vol. 1. Heriot-Watt University. 2003. pp. 5-11.
3. S. Robertson, J. Robertson. *Mastering the Requirements Process*. 3th ed. Ed. Addison Wesley. New Jersey, USA. 2006. pp. 48-79.
4. W. Serrano. “¿Qué constituye a los lenguajes natural y matemático?”. *SAPIENS: Revista Universitaria de Investigación*. Vol. 6. 2005. pp. 47-59.
5. R. Vernengo. “El discurso del derecho y el lenguaje normativo”. *Revista ISONOMÍA*. Vol. 4. 1996. pp. 87-95.
6. D. Berry. “Natural language and requirements engineering - Nu?”. *International Workshop on Requirements Engineering*. Vol. 1. 2003. pp. 1-7.
7. Tendales. “Lógica simbólica. Lógica proposicional”. *Curso de Filosofía*. Ed. Educastur. Asturias, España. Disponible en: <http://blog.educastur.es/tendales/files/2009/12/logica-teoria2.pdf>. Consultado en mayo de 2011
8. J. Richard, H. Wojcik, J. Hoard. “Controlled Languages in Industry”. *Survey of the State of the Art in Human Language Technology*. NSF & European Commission. Available: <http://www.cslu.ogi.edu/HLTsurvey/ch7node8.html>. Accessed: May 2011
9. J. Haller, J. Schütz. *CLAT: Controlled Language Authoring Technology*. Proceedings of the 19th Annual international Conference on Computer Documentation. New York, USA. 2001. pp. 78-82.
10. L. Castro, F. Baiao, G. Guizzardi. “A survey on Conceptual Modeling from a Linguistic Point of View”. *Relatórios Técnicos do Departamento de Informática Aplicada da UNIRIO*. Vol. 019. 2009. pp. 3-12.
11. L. Araujo. “Procesamiento de Lenguaje Natural”. *Curso PLN Universidad Autónoma de México*. 2006. Disponible en: <http://tabasco.torreingenieria.unam.mx/gch/PLN/cap1.pdf>. Consultado en mayo de 2011.
12. A. Clegg. *Computational-Linguistic Approaches to Biological Text Mining*. PhD Thesis. Escuela de Cristalografía, University of London. London, England. 2008. pp. 41-60.
13. H. Cunningham. *Software Architecture for Language Engineering*. PhD Tesis. Departamento de ciencias de la computación, University of Sheffield. Sheffield, England. 2000. pp. 205-209.
14. A. Gangopadhyay. “Conceptual Modeling from Natural Language Functional Specifications”. *Artificial Intelligence in Engineering*. Vol. 15. 2001. pp. 207-218.
15. C. Zapata, F. Villa. “La Gramática Básica de UNLencep expresada en HPSG”. *Revista Avances en Sistemas e Informática*. Vol. 5. 2008. pp. 81-92.
16. M. Christel, K. Kang. *Issues in Requirements elicitation*. Technical Report CMU/SEI-92-TR-012 ESC-TR-92-012. Software Engineering Institute. 1992. Available on: <http://www.sei.cmu.edu/library/abstracts/reports/92tr012.cfm>. Accessed: September 12, 2013.
17. C. Zapata, C. Palacio, N. Olaya. “UNC-ANALISTA: Hacia la Captura de un Corpus de Requisitos a Partir de la Aplicación del Experimento Mago de Oz”. *Revista EIA*. Vol. 7. 2008. pp. 25-40.
18. J. Leite. “A survey on requirements analysis”. *Advanced Software Engineering Project Technical Report RTP071*. Department of Information and Computer Science, University of California. California, USA. 1987. pp. 26-30.
19. C. Zapata. *Definición de un esquema preconceptual para la obtención automática de esquemas conceptuales de UML*. PhD Tesis. Universidad Nacional de Colombia. Medellín, Colombia. 2007. pp. 5-13.
20. J. Natt, B. Regnell, P. Carlshamre, M. Andersson, J. Karlsson. “Evaluating Automated Support for Requirements Similarity Analysis in Market-Driven Development”. *Requirements Engineering*. Vol. 7. 2001. pp. 20-33.
21. J. Natt, B. Regnell, V. Gervasi, S. Brinkkemper. “A Linguistic-Engineering Approach to Large-Scale

- Requirements Management”. *IEEE Software*. Vol. 22. 2005. pp. 32-39.
22. J. Natt, T. Thelin, B. Regnell. “An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development”. *Empirical Software Engineering*. Vol. 11. 2006. pp. 303-329.
 23. A. Post, I. Menzel, A. Podelski. “Applying restricted english grammar on automotive requirements: does it work? a case study”. Proceedings of the *REFSQ’11*. Vol. 1. 2011. pp. 166-180.
 24. B. Lee. *Automated Conversion from a Requirements Document to an Executable Formal Specification Using Two-Level Grammar and Contextual Natural Language Processing*. Ph.D. Dissertation. The University of Alabama at Birmingham. Alabama, USA. 2003. pp. 94-108.
 25. P. Jain, K. Verma, A. Kass, R. G. Vasquez. *Automated review of natural language requirements documents: generating useful warnings with user-extensible glossaries driving a simple state machine*. Proceedings of the 2nd India software engineering conference (ISEC ‘09). Pune, India. 2009. pp. 37-46.
 26. L. Mich. “NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA”. *Nat. Lang. Eng.* Vol. 2. 1996. 161-187.
 27. V. Ambriola, V. Gervasi. “On the Systematic Analysis of Natural Language Requirements with CIRCE”. *Automated Software Engg.* Vol. 13. 2006. pp. 107-167.
 28. H. Yang, A. Willis, A. De Roeck, B. Nuseibeh. *Automatic detection of noxious coordination ambiguities in natural language requirements*. Proceedings of the IEEE/ACM international conference on Automated software engineering (ASE ‘10). California, USA. 2010. pp. 53-62.
 29. A. Sampaio, A. Rashid, R. Chitchyan, P. Rayson. “EA-Miner: towards automation in aspect-oriented requirements engineering”. *Transactions on aspect-oriented software development III*. Vol. 46. 2007. pp. 4-39.
 30. S. Nanduri, S. Rugaber. *Requirements validation via automated natural language parsing*. Proceedings of the 28th Hawaii International Conference on System Sciences. Hawaii, USA. 1995. pp. 362-368.
 31. S. Spreeuwenberg, K. Anderson. SBVR’s approach to controlled natural language. *Lectures Notes in Computer Science*. Vol. 5972. 2010. pp. 155-169.
 32. I. Bajwa, M. Lee, B. Bordbar. *SBVR Business Rules Generation from Natural Language Specification*. Proceedings of the 2011 AAAI Spring Symposium Series. California, USA. 2010. pp. 2-8.
 33. R. Cavada, A. Cimatti, A. Mariotti, C. Mattarei, A. Micheli, S. Mover, M. Pensallorto, M. Roveri, A. Susi, S. Tonetta. *Supporting Requirements Validation: The EuRailCheck Tool*. Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering (ASE ‘09). Auckland, New Zealand. 2009. pp. 665-667.
 34. O. Ormandjieva, I. Hussain, L. Kosseim. *Toward a text classification system for the quality assessment of software requirements written in natural language*. 4th international workshop on Software quality assurance: in conjunction with the 6th ESEC/FSE joint meeting (SOQUA ‘07). Dubrovnik, Croatia. 2007. pp. 39-45.
 35. C. Zapata, A. Gelbukh, F. Arango. “A Novel CASE Tool based on Pre-Conceptual Schemas for Automatically Obtaining UML Diagrams”, *Proc. of Revista Avances en Sistemas e Informática*. Vol. 4. 2007. pp. 117-124.
 36. C. Zapata. *Computational linguistics for helping requirements elicitation: a dream about automated software development*. Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas (YIWCALA ‘10). Los Angeles, USA. 2010. pp. 117-124.
 37. D. de Almeida, A. Rodrigues. *A requirements specification case study with ProjectIT-studio/requirements*. Proceedings of the 2008 ACM symposium on Applied computing (SAC ‘08). Ceará, Brazil. 2008. pp. 656-657.
 38. H. Dalianis. *Aggregation in the NL-generator of the Visual and Natural language Specification Tool*. Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics (EACL ‘95). Zaragoza, España. 1995. pp. 286-290.
 39. R. Schwitter, N. Fuchs. *Attempto - From Specifications in Controlled Natural Language towards Executable Specifications*. Proceedings of the 3rd Knowledge Acquisition for Knowledge-Based Systems Workshop. Toulouse, Francia. Vol. 15. 1996. pp. 377-394.
 40. G. Cabral, A. Sampaio. “Formal Specification Generation from Requirement Documents”. *Electron. Notes Theor. Comput. Sci.* Vol. 195. 2008. pp. 171-188.
 41. T. de Sousa, J. Almeida, S. Viana, J. Pavón. “Automatic analysis of requirements consistency with the B

- method”. *SIGSOFT Softw. Eng. Notes*. Vol. 35. 2010. pp. 1-4.
42. E. Reiter, R. Dale. “Building applied natural language generation systems”. *Nat. Lang. Eng.* Vol. 3. 1997. pp. 57-87.
 43. D. de Almeida, A. Rodrigues. *A Controlled Natural Language Approach for Integrating Requirements and Model-Driven Engineering*. Proceedings of the 2009 4th International Conference on Software Engineering Advances (ICSEA '09). Porto, Portugal. 2009. pp. 518-523.
 44. M. Gordon, D. Harel. *Generating Executable Scenarios from Natural Language*. Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '09). México DF., México. 2009. pp. 456-467.
 45. M. Elbendak, P. Vickers, N. Rossiter. “Parsed use case descriptions as a basis for object-oriented class model generation”. *J. Syst. Softw.* Vol. 84. 2011. pp. 1209-1223.
 46. C. Videira, A. Rodrigues. *Patterns and metamodel for a natural-language-based requirements specification language*. Proceedings of the CaiSE'05 Forum. Porto, Portugal. 2005. pp. 189-194.
 47. K. Hinge, A. Ghose, G. Koliadis. *Process SEER: A Tool for Semantic Effect Annotation of Business Process Models*. Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference (EDOC '09). Auckland. New Zealand. 2009. pp. 54-63.
 48. M. Osborne, C. MacNish. *Processing Natural Language Software Requirement Specifications*. Proceedings of the 2nd International Conference on Requirements Engineering (ICRE '96). 1996. pp. 229.
 49. V. Gervasi, D. Zowghi. “Reasoning about inconsistencies in natural language requirements”. *ACM Trans. Softw. Eng. Methodol.* Vol. 14. 2005. pp. 277-330.
 50. D. Popescu, S. Rugaber, N. Medvidovic, D. Berry. “Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models”. *Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs*. Vol. 53. 2008. pp. 103-124.
 51. T. Breaux, A. Antón, J. Doyle. “Semantic parameterization: A process for modeling domain descriptions”. *ACM Trans. Softw. Eng. Methodol.* Vol. 18. 2008. pp. 27.
 52. D. Deeptimahanti, R. Sanyal. *Semi-automatic generation of UML models from natural language requirements*. Proceedings of the 4th India Software Engineering Conference (ISEC '11). Thiruvananthapuram, India. 2011. pp. 165-174.
 53. M. Sullabi, Z. Shukur. “SNL2Z: Tool for Translating an Informal Structured Software Specification into Formal Specification”. *American Journal of Applied Science*. Vol. 5. 2008. pp. 378-384.
 54. D. Almeida, A. Rodrigues. *The ProjectIT-Studio/Requirements Case Tool - A Practical Requirements Specification Case Study*. In Proceedings of the CISTI 2008 /3^a Conferencia Ibérica de Sistemas y Tecnologías de la Información. Ourense, España. 2008. pp. 12.
 55. Y. Ko, S. Park, J. Seo, S. Choi. “Using classification techniques for informal requirements in the requirements analysis-supporting system”. *Inf. Software Technology*. Vol. 49. 2007. pp. 1128-1140.
 56. R. Lecoeuche, C. Mellish, D. Robertson. *A Framework for Requirements Elicitation through Mixed-Initiative Dialogue*. Proceedings of the 3rd International conference on Requirements Engineering: Putting Requirements Engineering to Practice (ICRE '98). Colorado Springs, USA. 1998. pp. 190.
 57. W. Black. *Acquisition of conceptual data models from natural language descriptions*. Proceedings of the third conference on European chapter of the Association for Computational Linguistics (EACL '87). Copenhagen, Denmark. 1987. pp. 241-248.
 58. G. Fliedl, C. Kop, H. Mayr, A. Salbrechter, J. Vohringer, G. Weber, C. Winkler. “Deriving static and dynamic concepts from software requirements using sophisticated tagging”. *Data Knowl. Eng.* Vol. 61. 2007. pp. 433-448.
 59. C. Zapata, A. Gelbukh, F. Arango. “Pre-conceptual Schema: a UML Isomorphism for Automatically Obtaining UML Conceptual Schemas”. *Advances in Computer Science and Engineering*, Vol. 19. 2006. pp. 3-13.