

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220136683>

Análisis de un Caso de Estudio en KCPM para la Generación de Diagramas de Clases.

Article · January 2005

Source: DBLP

CITATION

1

READS

67

3 authors:



[Carlos Mario Zapata](#)

National University of Colombia

137 PUBLICATIONS 221 CITATIONS

SEE PROFILE



[Aldrin Jaramillo Franco](#)

University of Antioquia

8 PUBLICATIONS 11 CITATIONS

SEE PROFILE



[Fernando Arango Isaza](#)

National University of Colombia

32 PUBLICATIONS 90 CITATIONS

SEE PROFILE

All content following this page was uploaded by [Carlos Mario Zapata](#) on 21 March 2017.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Zapata, Carlos Mario; Arango, Fernando; Jaramillo, Aldrin Fredy

Análisis de un Caso de Estudio en KCPM para la Generación de Diagramas de Clases
Avances en Sistemas e Informática, vol. 2, núm. 1, junio, 2005, pp. 27-39
Universidad Nacional de Colombia
Colombia

Disponible en: <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=133115025004>

Revista
**Avances en
Sistemas e Informática**
Escuela de INGENIERÍA de SISTEMAS
Facultad de Minas

Avances en Sistemas e Informática
ISSN (Versión impresa): 1657-7663
mprada@unalmed.edu.co; avances@unalmed.edu.co
u.co
Universidad Nacional de Colombia
Colombia

¿Cómo citar?

Número completo

Más información del artículo

Página de la revista

Análisis de un Caso de Estudio en KCPM para la Generación de Diagramas de Clases

Carlos Mario Zapata y Fernando Arango

UNIVERSIDAD NACIONAL DE COLOMBIA. Facultad de Minas.

*Escuela de Sistemas. Grupo UN-INFO
{cmzapata ; farango}@unalmed.edu.co*

Aldrin Fredy Jaramillo

UNIVERSIDAD DE ANTIOQUIA.

*Departamento de Ingeniería de Sistemas
afjara@udea.edu.co*

Recibido para revisión Mar 2005, aceptado Abr 2005, versión final recibida Abr 2005

Resumen: La generación semiautomática de esquemas conceptuales (especialmente diagramas de clases) a partir de modelos verbales, ha sido abordada con alguna profundidad para idiomas como el inglés, francés y alemán, entre otros; no obstante, se han realizado pocas propuestas para el idioma español son sólo incipientes. En este artículo se discute la generación del diagrama de clases usando el método KCPM (Klagenfurt Conceptual Predesign Model) a partir de un modelo verbal en idioma español, el cual se aplica a un caso de estudio; se propone adicionalmente la utilización de árboles sintácticos clásicos, para mejorar la legibilidad del análisis sintáctico, y la adición de una columna en el esquema KCPM para incluir el rol temático asumido, para facilitar la aplicación de las reglas de conversión. Los resultados obtenidos muestran las bondades de las modificaciones propuestas, e incentivan la realización de nuevas investigaciones sobre esta temática.

Palabras Clave: Diagrama de Clases, KCPM, generación semiautomática de esquemas conceptuales

Abstract: There are few and incipient proposals about semi-automatic generation of conceptual schemas (specially class diagram) from Spanish verbal models. However, for other languages like English, French and German, among others, there are more experience. In this paper, we discuss class diagram generation based on KCPM (Klagenfurt Conceptual Predesign Model), from Spanish, and we apply it to a case study; furthermore, we propose the use of classic syntactic trees, for syntactic analysis legibility purposes, and we add an additional column to KCPM schema for assumed thematic role inclusion, for facilitating conversion rules applying. Obtained results show the goodness of proposed modifications, and promote new research about this topic.

Keywords: Class Diagram, KCPM, semiautomatic generation of conceptual schemas

1 INTRODUCCIÓN

Los esquemas conceptuales para el diseño de software (e.j. Modelo Entidad-Relación, diagramas de clases UML, etc.) buscan describir formalmente los requerimientos de un sistema, de una manera clara y estructurada, a partir de una sintaxis y semántica que no genere ambigüedades en su interpretación. Sin embargo, los clientes y usuarios de los sistemas no están usualmente en la capacidad de expresar sus requisitos en dichos esquemas; de tal forma, que las descripciones en lenguaje natural (LN) son el insumo utilizado por el Ingeniero de

Software en las etapas tempranas del proceso de Ingeniería de Requisitos [[Achour y Rolland \(1997\)](#)].

Denger, Dörr y Kamsties (2001) presentan una revisión del estado del arte en la generación semiautomática de modelos para diseño de software a partir del LN, en la cual se encuentran propuestas basadas en los idiomas Inglés, Francés y Alemán, pero no se presentan propuestas en lenguaje Español. Entre estas propuestas aparece KCPM (Klagenfurt Conceptual Predesign Model) como una de las propuestas más completas para el Alemán y el Inglés [[Mayr y Kop \(2002\)](#)], pero la cual presenta algunos problemas especialmente

de legibilidad del análisis sintáctico y selección de los roles temáticos de las palabras.

En este artículo se discute la aplicación de KCPM a modelos verbales en idioma Español para la generación automática del diagrama de clases y se realiza el análisis de un caso de estudio basado en el dicho método KCPM. A partir del análisis se proponen mejoras al mismo, como la sustitución de los árboles sintácticos de KCPM, que presentan dificultades de legibilidad, por árboles sintácticos clásicos; además, se incluye una nueva columna en la tabla de KCPM para contribuir en la aplicación de las reglas de mapeo hacia el diagrama de clases.

Este artículo está organizado de la siguiente manera: en la sección 2 se describe la arquitectura de KCPM explicando las características de los analizadores sintáctico, semántico y el intérprete. En la sección 3, mediante un caso de estudio, se ilustra el método y se examinan los principales hallazgos. Las secciones 4 y 5 están dedicadas a la presentación de conclusiones y trabajos futuros respectivamente.

2 ARQUITECTURA DE KCPM

El objetivo del método KCPM [Mayr y Kop (2002)] es generar semi-automáticamente modelos conceptuales, a partir de modelos verbales escritos en una forma de alemán simple, limitado sólo a frases afirmativas y condicionales simples, generalmente con un sujeto único y pocos predicados. Su arquitectura, presentada en la Figura 1, posee tres componentes funcionalmente diferenciables en el proceso, los cuales realizan las tareas de análisis sintáctico, análisis semántico e interpretación. Cada uno de ellos requiere recursos computacionales que suministran información del dominio y el lenguaje. Estos elementos se describen en detalle en las siguientes secciones.

2.1 Analizador Sintáctico

Formalmente una lengua es un conjunto de oraciones, donde cada oración es una cadena de símbolos pertenecientes al vocabulario de la lengua. Desde esta perspectiva, una gramática es una especificación formal y finita de este conjunto de oraciones. Esta especificación puede ser llevada a cabo mediante enumeración si el conjunto de oraciones es finito; si el conjunto de oraciones es infinito o el número de posibles oraciones es demasiado elevado, resulta imposible o poco práctico describirlas por enumeración. Alternativamente, una gramática es un mecanismo mediante el cual, con un determinado conjunto de reglas, es posible establecer si una oración pertenece o no al lenguaje descrito por la gramática. Esta gramática suele ser una gramática de estructura de frase o gramática sintagmática (PSG: *Phrase Structure*

Grammar) [Hopcroft y Ullman (1979)], la cual ha sido y es ampliamente utilizada en aplicaciones de LN, ya que se caracteriza por aportar una especificación formal de una lengua, lo cual facilita su implementación mediante algoritmos computacionales. Esta implementación, es decir, el algoritmo para determinar si la frase está bien escrita se denomina *parser*.

La tarea de análisis sintáctico también es denominada análisis superficial e incluye tanto el análisis léxico (es decir el análisis interno de la estructura de las palabras), como el análisis de la estructura de las frases como conjuntos de palabras. En este proceso se recibe como entrada el modelo verbal y como salida se tienen los árboles sintácticos que corresponden a las sentencias gramaticales, es decir, sentencias que pertenecen a la gramática. Pueden existir sentencias no gramaticales, que no sean reconocidas por el analizador sintáctico, en cuyo caso la salida del proceso corresponderá a un mensaje advirtiendo la situación para que el interesado corrija las frases de entrada. De acuerdo con Ortiz (2000) este proceso contempla dos modos diferentes:

- Análisis de constituyentes o análisis de la estructura de la frase, es decir, la estructuración de la frase en sus partes constituyentes y la categorización de éstas como nominales, verbales, adjetivales, etc.
- Análisis de las relaciones o funciones gramaticales, tales como sujeto, objeto, etc.

Para el análisis sintáctico, KCPM usa la denominada Morfosintaxis Teórica del Lenguaje Natural (NTMS por sus siglas en inglés); sin embargo, en este artículo, para efectos de claridad, se ejemplificará el proceso de análisis sintáctico con una gramática equivalente en lenguaje español, que tiene el mismo efecto final sobre el análisis.

Una *Phrase Structure Grammar* tiene cuatro componentes:

1. Un vocabulario terminal T : Las palabras o símbolos de la lengua definida.
2. Un vocabulario no terminal N : Los símbolos que no pertenecen al anterior y que se usan para especificar la gramática. Se define V como la unión de estos dos vocabularios ($V = T \cup N$).
3. Un conjunto de reglas o producciones P , donde cada regla es de la forma $a \geq b$ donde a es una secuencia de uno o más símbolos de V (es decir, $a \in V^+$) y b es una secuencia de cero o más símbolos de V ($b \in V^*$).
4. El símbolo inicial S , miembro de N .

La operación básica de una PSG es la de reescritura, es decir, la conversión de una secuencia de símbolos LHS (*Left Hand Side*) en otra diferente (RHS: *Right Hand*

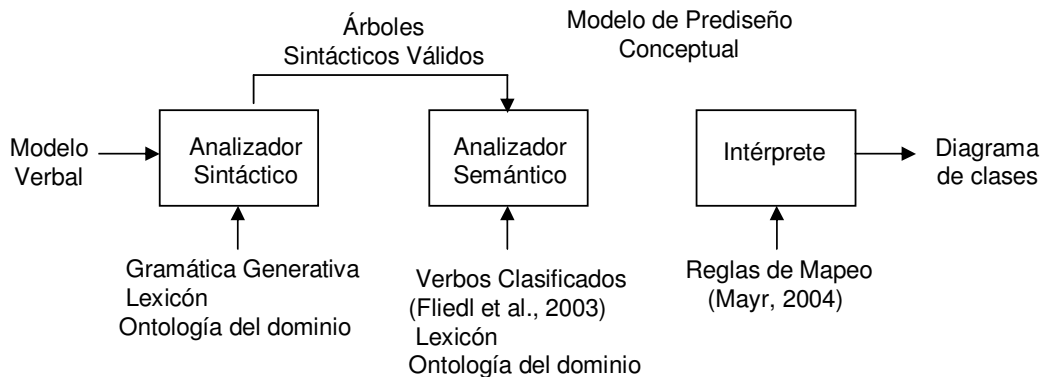


Figura 1: Arquitectura del Método KCPM para la Generación de Diagramas de Clases a partir de Modelos Verbales Simples

Side). Una sencilla gramática para un subconjunto del español podría estar determinada por las reglas que se muestran en la Figura 2. En estas reglas S es la oración, SN el sintagma nominal, SV el sintagma verbal, Npr el nombre propio, DET el determinante, N el sustantivo y V el verbo. Algunas oraciones que corresponden a esta gramática son: *Juan limpió* (Reglas: 1, 2, 3, 5, 7) y *Juan limpió el computador* (Reglas: 1, 2, 5, 3, 7, 2, 4, 6), cuyos árboles sintácticos se muestran en la Figura 3.

En general, el problema de la ambigüedad se elude en la propuesta de Klagenfurt mediante el uso de frases afirmativas con un solo sujeto y pocos objetos, evitando adjetivos y frases condicionales de la forma *si - entonces*, con frases afirmativas simples en su interior.

2.2 Analizador Semántico

Este componente recibe como entrada los árboles sintácticos elaborados en la etapa anterior, en los cuales se establecen las categorías de los lexemas y arroja como resultado la clasificación semántica de los constituyentes de la sentencia y un conjunto de tablas para el problema específico que constituyen el Modelo de Prediseño Conceptual del método de Klagenfurt. El proceso de análisis semántico es conocido como análisis profundo puesto que busca establecer el sentido de cada uno de los elementos de la frase como unidad y en contexto.

El método KCPM está basado en la gramática de casos planteada por Fillmore (1968). En esta teoría se reconoce que, aparte de las relaciones sintácticas y funcionales, en una oración existen relaciones semánticas entre el verbo y otras partes de la oración que se denominan “Roles Temáticos” o “Casos Profundos”. En la Gramática de Casos, el verbo es el foco de una oración, y las sentencias nominales en ésta se encuentran en una relación de rol específico con el verbo. Esta gramática clasifica cada verbo de acuerdo con su Marco de Casos y algunos aspectos a resaltar de ella son:

- Se ha asumido empíricamente que un pequeño conjunto de roles temáticos es suficiente para modelar

todos los verbos en el lenguaje.

- Para cada rol existe alguna estructura en algún lenguaje que trata el rol de manera distinta.
- Cada verbo posee roles temáticos obligatorios y opcionales.
- Un cierto rol temático ocurre en una oración sólo una vez.

Los roles temáticos definidos por Fillmore (1968) son:

- Agentivo (quién): El agente animado de una acción.
- Instrumental (cómo): Involucra una fuerza inanimada o un objeto.
- Dativo (quién): Participante animado que se ve afectado por una acción.
- Factitivo (qué): El objeto resultante de una acción.
- Locativo (cuándo, dónde): Posición local de un estado o de una actividad.
- Objetivo (todo lo demás): Objetos inanimados afectados.

KCPM se basa también en los roles temáticos propuestos por Gruber (1965), a los cuales denominan “Roles Theta”:

- Agente: Instigador animado de una acción.
- Experimentador: Reconocedor animado
- Tema: Entidad que sufre un cambio de estado / localización (paciente).
- Estímulo: De un estado o evento mental / perceptual.
- Receptor: De una transferencia.

Regla 1.	S -> SN, SV	Regla 2.	SN -> Npr iDET, N
Regla 3.	SV -> V V, SN	Regla 4.	DET -> el un
Regla 5.	Npr -> Juan	Regla 6.	N -> computador
Regla 7.	V -> limpió imprimió		

Figura 2: Reglas gramáticas sencillas para un subconjunto del idioma Español

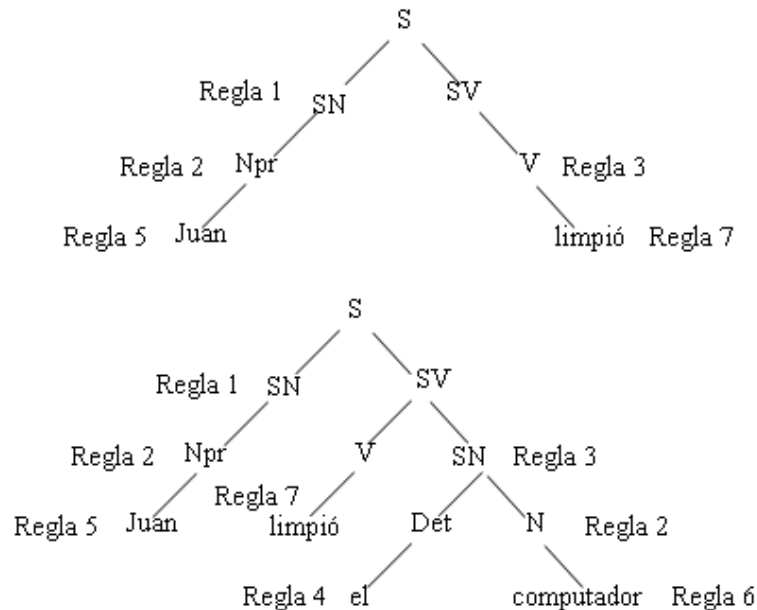


Figura 3: Árboles gramaticales de las frases reconocidas por el analizador sintáctico

- Beneficiario: Para aquellos que se benefician de una acción que se toma.

Para la determinación de los roles temáticos se emplean los lexicones. Un lexicon es un diccionario que contiene las categorías gramaticales asociadas con una palabra y, dependiendo de qué tan especializado sea, puede tener la estructura de roles temáticos para los verbos y otros elementos para el procesamiento del lenguaje natural [Dorr (2001)].

Un verbo puede presentar diferentes roles temáticos según el uso o significado que presente. Por ejemplo, en la frase “el hombre inteligente sabe del tópico”, existen dos posibles significados del verbo saber: i) Conocer y ii) Gustar. En KCPM se elige el significado con base en los roles temáticos que acompañan el verbo, aunque en una propuesta de mejoramiento podrían emplearse métodos estadísticos u ontológicos para determinar ese significado. En este caso en particular, el hecho de que “tópico” desempeña el rol temático de “Tema” en relación con el verbo saber y que está restringido a Información, permite deducir que el sentido del verbo es “conocer”. Luego de ser determinado el sentido de utilización del verbo, se procede a identificar los roles temáticos definidos en el lexicon y a etiquetar los constituyentes de la frase de acuerdo con el rol temático correspondiente a que tu-

viesen lugar.

Una vez establecidas las categorías sintácticas y semánticas de los constituyentes, se realiza la identificación de los elementos del Modelo de Prediseño Conceptual para el diagrama de Clases de UML. Bajo el enfoque propuesto por Mayr y Kop (2002), los principales conceptos son:

- Tipo cosa: Generalización de la noción de clase de UML (Incluye clases y atributos)
- Tipos conexión: Generalización del concepto asociación de UML.
- Perspectiva: Punto de vista o aspecto de un Tipo conexión. Por ejemplo, comprar y vender pueden ser dos aspectos de un mismo Tipo-conexión.
- Determinante de la perspectiva: Es sinónimo del concepto de Rol utilizado en UML.
- Determinante de tipo conexión: Es la utilización de las formas sustantiva y verbal de una palabra. Por ejemplo, hablar de “factura” en una frase y de “facturar” en otra.

En KCPM el objetivo consiste en llenar un conjunto de tablas, identificando las palabras que se pueden ubicar

en los elementos descritos; se deja para la siguiente etapa la definición de la categoría UML a la cual pertenecen. Por ejemplo, al agrupar en la categoría tipo cosa a todos los sustantivos identificados, corresponde a la siguiente etapa del proceso establecer si cada uno de ellos hace referencia a una clase o a un atributo. El modelo KCPM aplicado a la conversión al diagrama de clases incluye dos tablas: una para los tipos cosa y otra para los tipos conexión. La tabla de los tipos cosa incluye las siguientes columnas:

- Identificación: es un código que se le asigna al tipo cosa para diferenciarlo de manera única a lo largo del proceso.
- Nombre.
- Descripción cuantitativa (cuando la hay en el texto). Por ejemplo, cuando se dice “en la planta hay 200 empleados”, 200 es una descripción cuantitativa.
- Ejemplos, que son aclaraciones como nombres propios o listas de objetos, como por ejemplo “El Señor Pérez es un accionista” o “una solicitud puede ser aprobada o reprobada”; son ejemplos en estas frases “El Señor Pérez” y “aprobada” o “reprobada”.
- Valores del dominio: son posibles rangos de valores que se pueden presentar en el modelo verbal. Por ejemplo, cuando se dice “los profesores deben tener una edad inferior a 30 años”, “inferior a 30 años” es un valor del dominio del tipo cosa “edad”.
- Sinónimo: cuando hay varios tipos cosa que tienen el mismo significado para el dominio que se aplica.
- Descripción: es una definición opcional que se incluye en la tabla para mayor claridad del tipo cosa al que se refiere.
- Fuente del requisito: es el número de la frase correspondiente al modelo verbal en que se encuentra ubicado el tipo cosa.

Para los tipos conexión, las columnas de la tabla son las siguientes:

- Identificación: es un código que identifica de manera única el tipo conexión.
- Nombre
- Determinante tipo conexión
- Perspectiva. Tiene las siguientes subcolumnas:
 - Identificación de la perspectiva.
 - Tipo cosa involucrado.
 - Nombre de la perspectiva.

- Valores mínimos y máximos de la perspectiva, cuando se establezcan explícitamente.
- Determinante de perspectiva.

- Fuente del requisito, que es el número de la frase correspondiente al modelo verbal en que se encuentra ubicado el tipo cosa.

2.3 Intérprete

El proceso de interpretación consiste en mapear, mediante la aplicación de un conjunto de reglas, los elementos del Modelo de Prediseño Conceptual a sus correspondientes elementos en el modelo conceptual, de manera particular al diagrama de clases.

2.3.1 El Diagrama de Clases

Es un diagrama estructural utilizado en el lenguaje unificado de Modelamiento UML [OMG (2004)] para agrupar y representar de manera gráfica los objetos del mundo mediante las “clases” que los agrupan. Las clases pueden poseer atributos, que son características representativas de las clases, y operaciones, que son acciones que se pueden realizar con las clases; igualmente, las clases pueden estar relacionadas entre sí con diferentes tipos de vínculos o relaciones. En la Figura 4 se compendia la simbología básica de un diagrama de clases.

2.3.2 Las Reglas de Mapeo

En Mayr y Kop (2002) se encuentra una versión de las reglas que incluye las siguientes:

Reglas para tipos cosas, leyes directas

Regla 1 a) (Ley) Un tipo cosa T es mapeado a una clase CT (a un tipo valor VT), si T es especificado como tal por el diseñador en la columna de clasificación.

Regla 1 b) (Ley) T es mapeado a un tipo valor VT, si T tiene una entrada en la columna valor del dominio.

Regla 2 (Ley) Un tipo cosa T es mapeado a una clase CT (a un tipo valor VT), si T está involucrado en un tipo conexión C mediante una perspectiva “es identificado por” (“identifica”).

Regla 3 (Ley) Un tipo cosa T es mapeado a una clase CT, si T es el único tipo cosa involucrado de un tipo conexión (Tipo conexión reflexiva; note que atributos reflexivos carecen de sentido).

Regla 4 (Ley) Un tipo cosa T es mapeado a una clase CT (una clase de asociación en particular), si T es un determinante de tipo conexión C1 y es un tipo

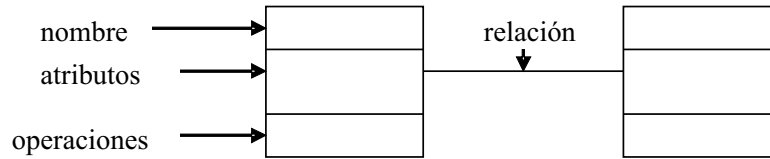


Figura 4: Simbología Básica de un Diagrama de Clases

cosa involucrado de otro tipo conexión C2 con C1 <> C2 (asociación convertida en objeto).

Regla 5 (Ley) Un tipo cosa T es mapeado a una clase CT, si T está involucrado en un tipo conexión C mediante una perspectiva la cual tiene un determinante de perspectiva.

Reglas para tipos cosas, leyes indirectas

Regla 6 (Ley) Un tipo cosa T es mapeado a una clase CT, si T está involucrado en un tipo conexión y todos los otros tipos de cosas involucrados ya han sido mapeados a tipos de valores. (Note: UML no suministra tipos de valores anidados).

Regla 7 (Ley) Un tipo cosa T es mapeado a una clase CT (a un tipo valor VT), si está conectado mediante un tipo conexión generalización o agregación a otro tipo cosa la cual ya ha sido mapeada a una clase (a un tipo valor).

Regla 8 (Propuesta) Un tipo cosa T puede ser mapeado a una clase CT, si T está involucrado en un tipo conexión A mediante una perspectiva con mínima cardinalidad 0.

Regla 9 (Propuesta) Un tipo cosa T puede ser mapeado a una clase CT, si T está involucrado en un tipo conexión C mediante una perspectiva cuyo nombre corresponde a un verbo agentivo (posiblemente predefinido en un lexicón).

Regla 10 (Propuesta) Un tipo cosa T puede ser mapeado a un tipo valor VT si su nombre es el segundo miembro de una composición (el primer miembro será mapeado a una clase)

Reglas para tipos cosas, propuestas indirectas

Regla 11 (Propuesta) Un tipo cosa T puede ser mapeado a una clase CT, si está conectado mediante tipos de conexión al menos a un tipo cosa, la cual ha sido previamente mapeada a una clase y al menos a un tipo cosa la cual ha sido previamente mapeada a un tipo valor.

Regla 12 (Propuesta) Un tipo cosa T puede ser mapeado a un tipo valor VT, si está involucrado en un solo tipo de conexión (por la mínima cardinalidad

1) y si no es un determinante de tipo conexión ni un determinante de perspectiva.

Reglas para tipos conexión, leyes indirectas

Regla 13 (Ley) Un tipo conexión C con tipos de cosas involucradas $T_1 \dots T_n$ es mapeado a una asociación si al menos dos de los tipos de cosas involucrados han sido mapeados a clases previamente.

Regla 14 (Ley) Un tipo conexión C con tipos de cosas involucrados $T_1 \dots T_n$ es mapeado a atributos $At_1 C \dots At_{i-1} C \dots At_{i+1} C \dots At_n C$, si únicamente uno sus tipos de cosas involucrados, T_i , fue mapeado a una clase CT_i mientras que todos los demás fueron mapeados a tipos de valores.

Nótese que las reglas anotadas no hacen alusión alguna a las operaciones que pueden hacer parte de una clase y también que algunas de las reglas enunciadas no tienen la fuerza de ley y por eso se colocan únicamente como propuestas; además, la regla 1 a) tiende a ser completamente manual, requiriendo un compromiso en la decisión por parte del analista, lo cual se pretende suprimir o minimizar con un método semi-automático. Estos aspectos pueden contribuir al mejoramiento y extensión de estas reglas con el fin de obtener mejores resultados en los diagramas de clases resultantes.

3 CASO ESTUDIO

Para efectos de este artículo se estudia y ejemplifica la generación del diagrama de clases - UML, aunque KCPM puede generar otros diagramas; además, se parte en el caso de estudio de un modelo verbal sencillo en español.

A continuación se presenta un ejemplo, empleado por Fliedl, Kop y Mayr (2003) para la obtención automática del diagrama de actividades (otro de los diagramas de UML [OMG (2004)]), pero que presenta ciertos inconvenientes para la construcción del diagrama de clases, como se verá más adelante:

Modelo Verbal:

- S1. La orden llega.
- S2. El departamento de pedidos verifica cada artículo en la orden.

- S3. Si cada artículo está disponible en el inventario, entonces el departamento de pedidos relaciona cada artículo en la orden.
- S4. Si la cantidad en inventario de un artículo es menor que la cantidad mínima, el encargado del inventario debe ordenar el artículo.
- S5. Si la orden llega el departamento de contabilidad verifica el pago.
- S6. Si el pago es autorizado y se tienen todos los artículos en inventario, entonces el departamento de pedidos libera la orden.
- S7. Si el pago es autorizado pero no se tienen todos los artículos en inventario, entonces el departamento de pedidos debe poner la orden con las órdenes pendientes.
- S8. Si el pago no es autorizado entonces el departamento de pedidos debe rechazar la orden.

Como insumo para este proceso se emplea el lexicón de la Tabla 1, el cual ha sido definido con base en Dorr (2001). Con el ánimo de mostrar el proceso sólo se tomarán las dos primeras sentencias.

Tabla 1: Lexicón utilizado como insumo para el proceso

Palabra	Categoría Gramatical	Roles Temáticos que puede asumir	Roles Temáticos que pueden ser acompañantes del verbo
Almacenista	Sustantivo (N)	agente, tema, fuente, experimentador	
Artículo	Sustantivo (N)	tema, experimentador	
Autorizado	Adjetivo (Adj)		
Cada	Adjetivo (Adj)		
Cantidad	Sustantivo (N)	tema, experimentador	
con	Preposición (Pr)		
Contabilidad	Sustantivo (N)	Agente, tema, fuente, localización, experimentador	
De	Preposición (Pr)		
Debe	Verbo auxiliar (V. aux)		Ayuda al verbo principal a asumir sus roles
departamento	Sustantivo (N)	agente, tema, fuente, localización, experimentador	
Disponible	Adjetivo (Adj)		
El	Artículo (Art)		
En	Preposición (Pr)		
Entonces	Adverbio (Adv)		
Es	Verbo (V).		Tema, predicado
Está	Verbo intransitivo (V. int)		Tema, predicado
Inventario	Sustantivo (N)	tema, experimentador	
la	Artículo (Art)		
Libera	Verbo transitivo (V. tr)		Agente, tema
Llega	Verbo intransitivo (V. int)		Tema, fuente, localización
Lo	Artículo (Art)		
Menor	Adjetivo (Adj)		
Mínima	Adjetivo (Adj)		
No	Adverbio (Adv)		
Orden	Sustantivo (N)	tema, experimentador	
ordenar	Verbo transitivo (V. tr)		Agente, tema, fuente
Pago	Sustantivo (N)	tema, experimentador	
Pedidos	Sustantivo (N)	tema, experimentador	
Continúa ...			

Tabla 1: Lexicón utilizado (continuación)

Palabra	Categoría Gramatical	Roles Temáticos que puede asumir	Roles Temáticos que pueden ser acompañantes del verbo
pendientes	Adjetivo (Adj)		
Pero	Conjunción (Conj)		
Poner	Verbo transitivo (V. tr)		Agente, tema, meta
Que	Pronombre (Ppr)		
Rechazar	Verbo transitivo (V. tr)		Experimentador, tema
Relaciona	Verbo transitivo (V. tr)		Agente, tema
Se	Pronombre (Ppr)		
Si	Conjunción (Conj)		
Tienen	Verbo transitivo (V. tr)		Tema, localización
Todos	Adjetivo (Adj)		
Un	Artículo (Art)		
Verifica	Verbo transitivo (V. tr)		Agente, tema, localización
Y	Conjunción (Conj)		

3.1 Análisis Sintáctico

En este caso, de acuerdo con las reglas de la gramática y el lexicón, para las dos frases iniciales del caso de estudio se obtienen las estructuras de constituyentes que se

muestran en la Figura 5. Obsérvese que, una vez finalizado el análisis sintáctico, se establecen las categorías gramaticales (artículos, preposiciones, sustantivos, etc.) de los constituyentes.

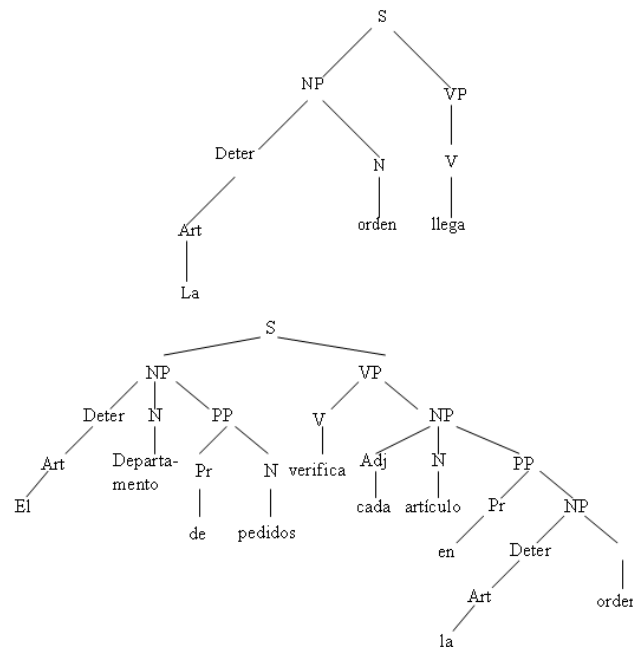


Figura 5: Árboles de Constituyentes de las dos primeras frases del Caso de Estudio

Es importante resaltar que las sentencias admitidas inicialmente obedecen a estructuras simples, en las cuales no se presentan problemas por efecto de ambigüedades o dobles interpretaciones. Estructuras complejas, tales como frases interrogativas o con la presencia de conjunciones, deberán ser preprocesadas con el

propósito de ser descompuestas en sentencias simples equivalentes, para luego efectuar el análisis de cada una de ellas. Otro problema que no se está resolviendo con el lexicón construido tiene que ver con el manejo de las conjugaciones (que no es tan grave en el lenguaje Alemán, pero sí en el ejemplo en Español) y el reconocimiento de

las diferentes formas de las palabras en el texto (sinonimia y plurales, por ejemplo).

3.2 Análisis Semántico

Tomando como base el lexicón, se obtienen los siguientes resultados:

Sentencia 1: La orden (tema) llega

Roles temáticos ausentes: fuente, localización

Sentencia 2: El Departamento de pedidos (agente) verifica cada artículo (tema) en la orden (localización)

Roles temáticos ausentes: Ninguno.

Nótese que “Departamento de pedidos” debe ser considerado como una unidad semántica, debido al significado de “pertenencia” que le entrega la preposición “de” a las dos palabras. En la Tabla 2 se presenta el Modelo de Prediseño Conceptual para los tipos cosa identificados del Caso de Estudio. A la descripción planteada por Mayr y Kop (2002) se adiciona la columna “Rol Asumido”, para incluir los resultados del análisis semántico para cada palabra dentro de las frases que participa. En la Tabla 3 se presenta el Modelo de Prediseño Conceptual para los tipos conexión que se identifican en el caso de estudio.

Tabla 2: Modelo de Prediseño Conceptual para los tipos cosa identificados en el Caso de Estudio

id#	Nombre	Descrip. cuantitativa	Ejem.	Valor dominio	Sinónimo	Descrip.	Fuente de requisitos	Rol asumido
D001	Orden						s1	Tema
							s2	Localización
							s3	Localización
							s5	Tema
							s6	Tema
							s7	Tema
							s8	Tema
D002	Departamento de Pedidos						s2	Agente
							s3	Agente
							s6	Agente
							s7	Agente
							s8	Agente
D003	Artículo	cada					s2	Tema
		cada					s3	Tema
							s3	Tema
							s4	Tema
							s4	Tema
		todos					s6	Tema
							s7	Tema
D004	Inventario						s3	Localización
		Mínima cantidad					s4	Localización
							s6	Localización
							s7	Localización
D005	Almacenista					s4	Agente	
D006	Departamento de contabilidad					s5	Agente	
D007	Pago						s5	Tema
							s6	Tema
							s7	Tema
							s8	Tema
D008	Pendientes					s7	Localización	
D009	Cantidad					s4	Tema	
Continúa ...								

Tabla 2: Modelo de Prediseño Conceptual para los tipos cosa (continuación)

id#	Nombre	Descrip. cuantitativa	Ejem.	Valor dominio	Sinónimo	Descrip.	Fuente de requisitos	Rol asumido
D010	Cantidad mínima						s4	Tema

Tabla 3: Modelo de Prediseño Conceptual para los tipos conexión identificados en el Caso de Estudio

c-id#	Nombre	Determ. Tipo Conex.	Perspectiva				Fuente Requis.
			persp#	Tipo cosa involucrado	Nombre	min/máx	
C001	Llega		p001	D001, orden	Llega		s1,s5
C002	Verifica		p002a	D002, Dep. pedidos	Verifica		s2
			p002b	D001, orden			s2
			p002c	D006, Dep. Contab.	Verifica		s5
			p002d	D007, Pago			s5
C003	Disponible		p003a	D003, Artículo	Disponible en		s3
			p003b	D004, Inventario			s3
C004	Relaciona		p004a	D002, Dep. pedidos	Relaciona		s3
			p004b	D003, Artículo			s3
			p004c	D001, orden			s3
C005	Es menor		p005a	D009, Cantidad	Menor que		s4
			p005b	D010, Mín cantidad			s4
C006	Ordenar	D001, orden	p006a	D005, Almacenista	Ordenar		s4
			p006b	D003, Artículo			s4
C007	Se autoriza		p007a	D007, Pago	Se autoriza		s6, s8
C008	Se tienen		p008a	D003, Artículo	Se tienen		s7
			p008b	D004, Inventario			s7
C009	Libera		p009a	D002, Dep. pedidos	Libera		s6
			p009b	D001, orden			s6
C010	Colocar		p010a	D002, Dep. pedidos	Colocar		s7
			p010b	D001, orden			s7
			p010c	D008, pendientes			s7
C011	Rechazar		p011a	D002, Dep. pedidos	Rechazar		s8
			p011b	D001, orden			s8

3.3 Intérprete

Teniendo como insumo las Tablas 2 y 3, resultantes de la etapa anterior, y aplicando las heurísticas presentadas en el apartado 2.3., en la tabla 4 se presentan los resultados de mapeo a elementos de UML para los tipos cosa y en la tabla 5 para los tipos conexión. Los elementos identificados se pueden apreciar en el diagrama de clases de la Figura 6.

De las Tablas 4 y 5, y de la Figura 6 se puede deducir lo siguiente:

- Sólo se utilizan dos reglas consideradas como “leyes” (regla 4 y regla 2) y tan sólo una “propuesta” (regla 9). Las demás leyes y propuestas aplicables a los tipos cosa no se utilizan.
- De los 10 tipos cosa definidos, sólo cinco pudieron ser mapeados a Clase y ninguno a Atributo. De las clases identificadas, sólo Orden y Pago provienen de leyes, es decir, en el intérprete basado en las reglas de ?) sólo esas son clases sin lugar a dudas, mientras que las demás identificadas podrían no ser clases.

Tabla 4: Resultados del Intérprete para los tipos Cosa

Regla	Sentencia	Tipo Cosa	Tipo de Elemento del Diagrama de Clases
R4	S1	Orden	Clase
R9	S2, S3, S6, S7, S8	Departamento de Pedidos	Clase
No aplicables	S2	Artículo	Ninguno
No aplicables	S6	Inventario	Ninguno
R9	S4	Almacenista	Clase
R9	S5	Departamento de Contabilidad	Clase
R2	S6, S7, S8	Pago	Clase
No aplicables	S7	Pendientes	Ninguno
No aplicables	S4	Cantidad	Ninguno
No aplicables	S4	Cantidad Mínima	Ninguno

Tabla 5: Resultados del Intérprete para los Tipos Conexión

Regla	Sentencia	Tipo Conexión	Tipo de Elemento del Diagrama de Clases
No aplicables	S1	Llega	Ninguno
R13	S2, S5	Verifica	Asociación
No aplicables	S3	Está disponible en	Ninguno
No aplicables	S3	Relaciona	Ninguno
No aplicables	S4	Es menor que	Ninguno
No aplicables	S4	Debe ordenar	Ninguno
No aplicables	S6, S8	Se autoriza	Ninguno
No aplicables	S4	Debe ordenar	Ninguno
R13	S6	Libera	Asociación
No aplicables	S7	Debe colocar	Ninguno
R13	S8	Debe rechazar	Asociación

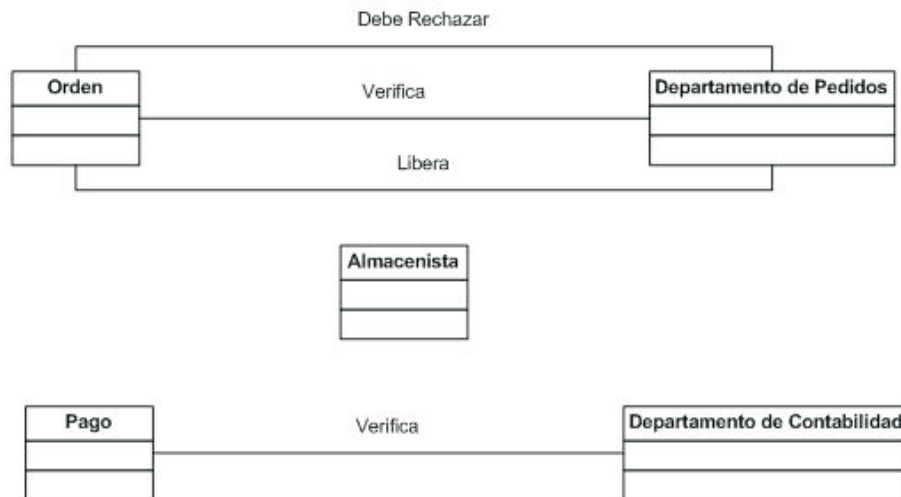


Figura 6: Diagrama de Clases Resultante

- De los 11 tipos conexión definidos, sólo tres pudieron ser mapeados a asociación.
- La regla 10 no es directamente aplicable al español, puesto que las composiciones de que trata son una práctica común para idiomas como el inglés y el alemán, pero no para el español.
- No se identificaron atributos en el modelo verbal, pese a que podría haber algunos candidatos como cantidad, cantidad mínima y pendientes.
- No existen reglas que permitan la determinación de operaciones para las clases, pese a que podrían existir algunos candidatos como verifica, relaciona, debe colocar o es menor que, entre otros.

Para que otras clases pudieran ser identificadas como tal empleando las reglas de ?, se deberían lanzar al usuario algunas preguntas que permitieran complementar la información con nuevas frases del modelo verbal, tales como:

- Una orden está compuesta de artículos?
- La cantidad y la cantidad mínima son características del Inventario?
- Cómo se diferencian los artículos?
- Cómo se liberan las órdenes?

Como se puede observar, existe una considerable diferencia en los resultados obtenidos frente a los deseados. Además, las heurísticas utilizadas no abordan el reconocimiento de las operaciones que hacen parte de las clases.

4 CONCLUSIONES

Se ha realizado un análisis del método KCPM para la generación semiautomática de diagramas de clases UML a partir de textos en lenguaje natural, un problema de gran relevancia en la literatura de la Ingeniería de Requisitos, pues la automatización de esta labor puede contribuir notablemente a la agilización de los procesos de desarrollo de software.

Se describió la arquitectura del método KCPM, el cual consta de un analizador sintáctico encargado de reconocer y categorizar sintácticamente los constituyentes de las frases, un analizador semántico responsable de la categorización de los roles temáticos desempeñados por los constituyentes en la frases y de la construcción del Modelo de Prediseño Conceptual y un intérprete encargado de llevar a cabo la generación del diagrama de clases mediante la aplicación de heurísticas sobre el Modelo de Prediseño Conceptual. En este artículo se realizaron dos

propuestas de mejoramiento al método KCPM, para facilitar la legibilidad del análisis sintáctico y la identificación de los roles temáticos asociados con las palabras, lo cual mejora la aplicabilidad del método.

Finalmente, el método y las limitaciones de las reglas de interpretación utilizadas por los investigadores de la Universidad de Klagenfurt se ilustraron con un caso de estudio, que permitió identificar algunas debilidades que podrían dar lugar a nuevos trabajos de investigación, tales como:

- Los textos en lenguaje natural que se deben ingresar para realizar la generación del diagrama son bastante simples y prácticamente inambiguos. El lenguaje natural es inherentemente ambiguo y este problema se viene tratando desde hace algunos años incluso en el lenguaje Español. Se nota en el método KCPM muy poco trabajo en el aspecto de resolución de la ambigüedad en diferentes etapas del proceso.
- El método tiene todavía una participación relativamente alta del analista. Hay ciertas claridades que debe realizar el analista a lo largo del proceso y ello se manifiesta por ejemplo en reglas de traducción como la primera en la cual, pese al automatismo que se pretende brindar al método, la ingerencia de los analistas y diseñadores se convierte en un asunto todavía irremplazable.
- Sólo se identifican parcialmente algunos elementos del diagrama de clases UML, aunque se podrían identificar otros, como por ejemplo las operaciones. Además, complementando la información correspondiente a los tipos cosa y a los tipos conexión, se podrían incluso generar otros diagramas UML adicionales.

5 TRABAJOS FUTUROS

Actualmente, bajo la asesoría del Grupo de Investigación en Ingeniería de Software de la Universidad Nacional de Colombia, se está trabajando en la construcción de un prototipo y la extensión de las reglas propuestas por los investigadores de Klagenfurt. Posteriormente, se plantearán trabajos en relación con temas diversos como:

- La asistencia al usuario para la correcta redacción de los modelos verbales. En esta línea de trabajo, se deberían tomar en consideración los trabajos tendientes a la resolución de las ambigüedades propias del lenguaje natural.
- Establecimiento de criterios de completitud, basados en roles temáticos y otras teorías equivalentes, para mejorar las posibilidades de generación de los diagramas.

- El mejoramiento de los recursos lingüísticos y los componentes del sistema de forma que se adapten mejor a las características particulares del idioma español.
- Generación de otros esquemas similares al Modelo de Prediseño Conceptual, que puedan contener información lingüísticamente valiosa para la generación de diagramas UML u otros tipos de diagrama empleados en la especificación de piezas de software.
- Depuración de las reglas heurísticas y su complementación para lograr trazar más elementos de UML para diversos diagramas.
- Definición de criterios de calidad para estas reglas y métodos de validación que partan de diagramas “correctos” y los comparen con diagramas generados.

REFERENCIAS

- Achour, C. y Rolland, C. (1997), Introducing genericity and modularity of textual scenario interpretation in the context of requirements engineering, Technical report, Centre de Recherche en Informatique, Université de Paris 1, Paris. En: CREWS Technical Report N°21.903.
- Denger, C., Dörr, J. y Kamsties, E. (2001), A survey on approaches for writing precise natural language requirements, Technical report, Fraunhofer IESE, Kaiserslautern. En: IESE-Report, N° 070.01.
- Dorr, B. (2001), Computational lexicon, Technical report, University of Maryland.
- Fillmore, C. (1968), *The Case for Case*, Holt, Rinehart and Winston, chapter Universals in Linguistic Theory.
- Fliedl, G., Kop, C. y Mayr, H. (2003), From scenarios to kcpm dynamic schemas: Aspects of automatic mapping, in ‘Natural Language Processing and Information Systems - NLDB’2003. Lecture Notes in Informatics P-29, GI-Edition, pp. 91–105.
- Gruber, J. (1965), *Studies in Lexical Relations*, PhD thesis, Cambridge, Mass: MIT.
- Hopcroft, J. y Ullman, J. (1979), *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley.
- Mayr, H. y Kop, C. (2002), A user centered approach to requirements modeling, in ‘Proc. Modellierung 2002. Lecture Notes in Informatics P-12 (LNI). GI-Edition, pp. 75–86.
- OMG (2004), ‘Object management group. unified modeling language uml’. En Línea: <<http://www.omg.org/uml>> C-03/05.
- Ortiz, A. (2000), ‘Estudios de lingüística española’. Facultad de Filosofía y Letras Universidad de Málaga. En Línea: <<http://elies.rediris.es/elies9/index.htm>> C-03/05.