

Caracterización ontológica de elementos básicos del diagrama de kaos a partir de lenguaje natural

Ontological characterization of basics of kaos chart from natural language

Luis Alfonso Lezcano-Rodríguez
M.Sc. Ing. Sistemas
Universidad Nacional de Colombia
Medellín, Colombia
lalezcan@unal.edu.co

Jaime Alberto Guzmán- Luna
Ph.D. Ingeniería
Universidad Nacional de Colombia
Medellín, Colombia
jaguzman@unal.edu.co

Resumen— En la ingeniería de software se utilizan diferentes tipos de diagramas para lograr la calidad que debe cumplir el sistema para desarrollar. El diagrama de objetivos de KAOS (Knowledge Acquisition in autOated Specification) es utilizado en las primeras fases del ciclo de vida de software (definición y análisis) para expresar a los interesados la importancia del sistema futuro. Sin embargo, en los trabajos que utilizan este diagrama no se logra identificar una automatización entre el lenguaje natural y los elementos básicos (objetivos, entidades, operaciones y agentes) que conforman dicho diagrama. En este artículo se propone la construcción de una ontología y la definición de un conjunto de reglas morfosintácticas y semánticas para: (i) caracterizar los elementos básicos a partir del uso del lenguaje natural en idioma español, (ii) minimizar la ambigüedad semántica presente en el universo del discurso, (iii) obtener automáticamente los elementos básicos, y (iv) elaborar automáticamente dicho diagrama.

Palabras clave— Ingeniería de software, procesamiento de lenguaje natural, educación de requisitos, validación de requisitos, ontologías, diagrama de objetivos de KAOS.

Abstract— In software engineering a variety of diagrams are used for achieving the quality that the system-to-be should accomplish. The KAOS (Knowledge Acquisition in autOated Specification) is used in the initial phases of the software life-cycle, for expressing to the stakeholders the importance of the future system. However, in the works that use this diagram, it is not identified an automation between natural language and the basic elements (goals, entities, operations and agents) that compose such diagram. In this article, the building of an ontology and the definition of a set of morphosyntactic and semantic rules are proposed for: (i) characterizing the basic elements based on their use in the Spanish natural language, (ii) minimizing the semantic ambiguity that exists in the domain of discourse, (iii) obtaining

automatically the basic elements, and (iv) building automatically KAOS goal diagram.

Keywords— Ingeniería de software, procesamiento de lenguaje natural, educación de requisitos, validación de requisitos, ontologías, diagrama de objetivos de KAOS.

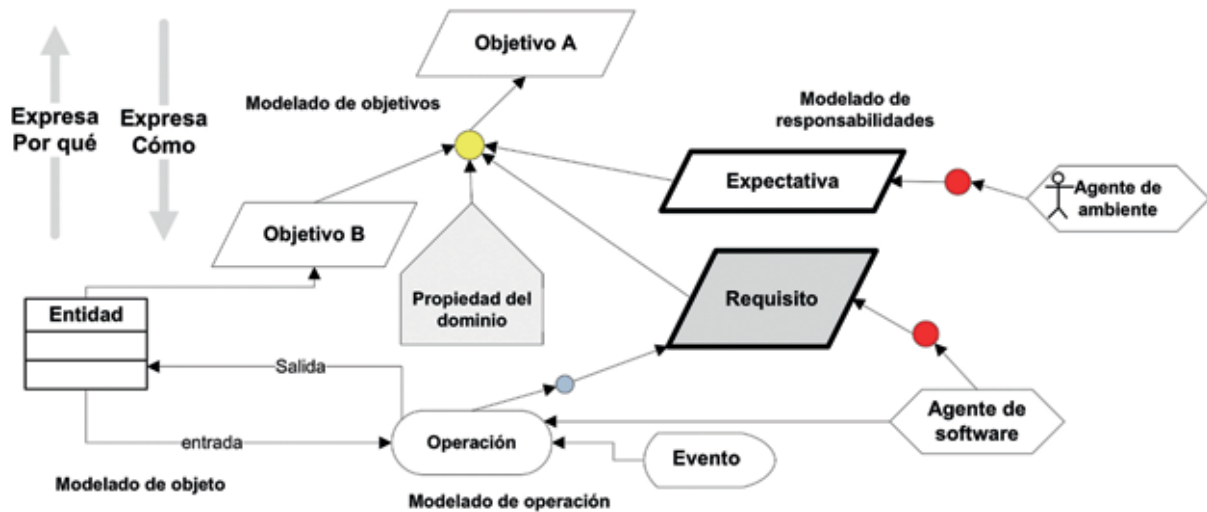
1. INTRODUCCIÓN

Alan D. [1] indica que las tareas correspondientes a la ingeniería de *software* inician antes de la construcción o de la elaboración del código, y continúan después de la finalización de la versión inicial del sistema (*software*).

Por otro lado, Ivar, Grady y James [10] señalan que en las primeras dos fases del ciclo de vida del *software* (definición y análisis) se desarrollan actividades para identificar, analizar y validar los requisitos que el sistema debe satisfacer. Sin embargo, las actividades asociadas a estas fases presentan problemas debido a la brecha de comunicación existente entre el analista de *software* (experto en lenguaje técnico) y el interesado (experto en terminología propia del área del problema).

Mientras Lamsweerde [12], [13] y [15] propone utilizar el diagrama de objetivos de KAOS para minimizar la brecha de comunicación vigente entre analista e interesado. Este diagrama incluye la diagramación de objetivos, entidades, operaciones, agentes, entre otros. En la Fig. 1 se puede observar un ejemplo con la simbología básica del diagrama de objetivos de KAOS.

Fig. 1. EJEMPLO CON LA SIMBOLOGÍA BÁSICA DEL DIAGRAMA DE OBJETIVOS DE KAOS



Fuente: Adaptado de [21].

En [2] se indica que la principal ventaja de la metodología KAOS está relacionada con la capacidad de alinear los requisitos, los objetivos y las expectativas de la organización. Sin embargo, en esta propuesta existe una alta participación del analista, durante la elaboración y construcción del diagrama de objetivos, es decir, es el responsable de interpretar e identificar, a partir del discurso del interesado (presentado en lenguaje natural), los elementos que componen el diagrama de objetivos. De hecho, Zapata, Villegas y Arango [25] señalan que actualmente es el analista quien a partir de la información suministrada por el interesado elabora el diagrama de objetivos de manera manual y con un alto grado de subjetividad sin que se definan ayudas para la conceptualización del diagrama obtenido. Por esto, como resultado del proceso técnico realizado por el analista, se diagraman elementos que, en la mayoría de los casos, no son consistentes con el discurso del interesado.

Con el fin de presentar una metodología que permita automatizar este proceso de obtención del diagrama de objetivos de KAOS, desde la expresión de requisitos en lenguaje natural, se presentan los siguientes estudios en el estado del arte: Si bien en [25] se utiliza el diagrama de objetivos durante el proceso de desarrollo de *software*, este no se orienta hacia la validación de los requisitos. Adicionalmente, la entrada al sistema propuesto está expresada en un lenguaje

controlado, que requiere una alta intervención del analista. Esto conlleva a que la expresividad se reduzca, de tal forma que a los requisitos les falta claridad y precisión.

Solucionando parcialmente estas debilidades, [24] presentan algunas estructuras lingüísticas para caracterizar objetivos. Sin embargo, su aplicación continúa siendo totalmente manual, realizada por el analista, lo cual no soluciona los problemas mencionados anteriormente, relacionados con la brecha en los vocabularios de analista y cliente. Además, las características de dichas reglas no facilitan su implementación, ya que requieren del análisis de clasificaciones de palabras.

Por su parte, Guzmán, Lezcano y Gómez [8] presentan una propuesta para caracterizar algunos de los elementos (objetivo, objetivo subrogado, requisito y agente) que conforman el diagrama de objetivos de KAOS. En este caso, no se presentan estructuras lingüísticas que permitan caracterizar los demás elementos (entidad, atributos, operaciones, entre otros) que conforman el diagrama de objetivos de KAOS. Asimismo, no se logra identificar en esta propuesta un procedimiento que permita minimizar la ambigüedad semántica de los elementos identificados. De manera que los problemas detectados en las propuestas de los autores que utilizan la metodología orientada a objetivos son: (i) no se observa una clara intervención del interesado durante el proceso identi-

cación y validación de requisitos mediante el uso del diagrama de objetivos de KAOS, (ii) no se logra identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos, (iii) no caracterizan u obtienen de manera automática los elementos básicos que conforman el diagrama de objetivos, y (iv) no se observa un procesamiento que logre minimizar la ambigüedad verbal presente en el universo del discurso.

Con el fin de mitigar estos problemas, se propone en este artículo una representación ontológica que garantiza: (i) la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos de KAOS (utilizado para la educación y validación de requisitos de *software*), (ii) la reducción de la ambigüedad semántica presente en el universo del discurso, (iii) la obtención automática de los elementos básicos (objetivos, entidades, operaciones y agentes), y (iv) la elaboración automática de dicho diagrama. Para ello, este artículo se organiza de la siguiente forma:

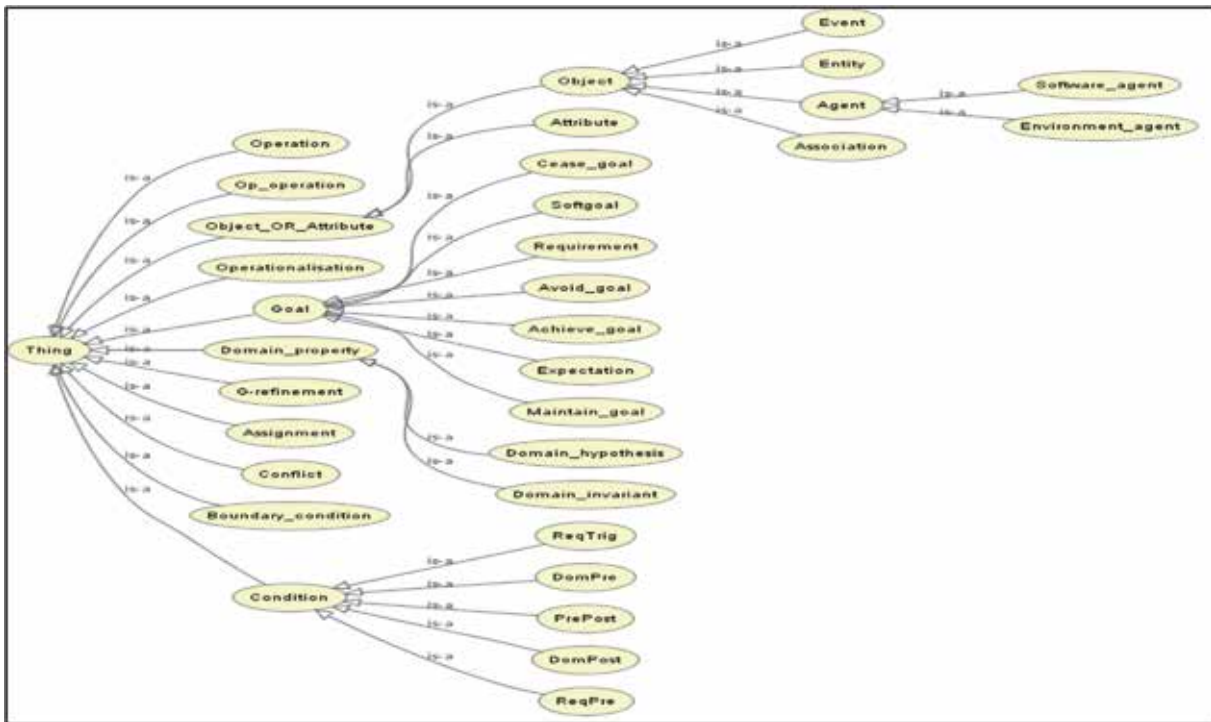
2. CONSTRUCCIÓN DE LA ONTOLOGIA KAOS

Para realizar una correcta definición y caracterización de los elementos del diagrama de objetivos de KAOS, se desarrolló una ontología que representa el metamodelo del diagrama de objetivos de KAOS. Para ello, se tomó como punto de partida el metamodelo propuesto por Matulevičius [18]. Asimismo, y haciendo uso de Protégé 4.1 se tomaron cada una de las clases que se exponen en dicho metamodelo, las cuales se modelaron teniendo en cuenta las generalizaciones existentes. Dado lo anterior, se logró obtener el modelo que se indica en la Fig. 2, que corresponde a la estructura de las clases.

Dado que el insumo de entrada es el lenguaje natural, es necesario definir los siguientes componentes gramaticales.

Perífrasis verbales: son construcciones sintácticas de dos o más verbos que funcionan como núcleo del predicado. Sirven para expresar las características de la acción verbal que no pueden señalarse mediante el uso de las formas sim-

Fig. 2. ESTRUCTURA DE LAS CLASES DEL METAMODELO DIAGRAMADO EN LA ONTOLOGÍA



Fuente: autores.

ples o compuestas [6]. Las perífrasis se pueden concentrar en dos grandes grupos, las perífrasis verbales aspectuales y las perífrasis verbales modales, estas últimas son las de real interés en esta propuesta, porque permiten enunciar acciones, tareas y deseos. Un ejemplo es “deber hacer”, ya que se presentan dos formas verbales, que informan sobre la intención del hablante. Pueden presentar un nexo entre los dos verbos, por ejemplo, en “tener que hacer”.

Sintagma nominal: es un grupo de palabras que se articulan alrededor del sustantivo, también llamado nombre, el cual corresponde al núcleo en este tipo de sintagma (Demonte y Bosque, 1999).

Verbo nominalizado: es aquel sustantivo proveniente de un verbo (ejemplo: realización, deriva de realizar).

Complemento: bajo este contexto es el conjunto de palabras que acompañan a la oración o forma de la oración de interés, con el fin de añadir sentido o significado.

Sujeto: corresponde a la persona, animal o cosa que realiza la acción del verbo [5].

Verbo no copulativo: un verbo copulativo es aquel que, junto con el atributo, forma el predicado nominal de una oración [5]. De manera que funcionan como enlace entre el sujeto y el atributo, por tanto, son verbos que carecen de significado, son casi vacíos, de hecho, son nominales, solo informan de un estado o un atributo, estos

son: ser, estar, parecer. Por el contrario, los no copulativos, son predicativos, así, dan indicaciones activas de lo que hace el sujeto. Dado lo anterior, se puede decir que el conjunto de los verbos que no sean copulativos (ser, estar, parecer), son no copulativos.

2.1 Reglas morfosintácticas para caracterizar objetivos

Los objetivos son declaraciones prescriptivas que el sistema debe satisfacer a través de la cooperación de sus agentes [14]. Las siguientes formas morfosintácticas que se proponen permitirán identificar en idioma español un objetivo:

FMOB1: *Sintagma nominal + Perífrasis verbal modal + Verbo nominalizado + Complemento.*

FMOB2: *Complemento A (opcional) + que + Sintagma nominal + Verbo + Complemento B.*

FMOB3: *Verbo en infinitivo + que + Sintagma nominal + Verbo + Complemento.*

FMOB4: *Sintagma nominal + Perífrasis verbal modal + Complemento.*

Asimismo, y haciendo uso de Protégé 4.1, se tomaron cada una de las clases que se exponen en dicho metamodelo, las cuales se modelaron considerando las generalizaciones existentes. Dado lo anterior, se logró obtener el modelo que se indica en la Fig. 2, que corresponde a la estructura de las clases.

TABLA I
EJEMPLO DE LAS FORMAS MORFOSINTÁCTICAS DEFINIDAS PARA CARACTERIZAR OBJETIVOS

Lenguaje natural	Forma utilizada	Elementos morfosintácticos identificados
El sistema debe permitir el registro de los clientes.	FMOB1	Sintagma nominal: “el sistema”. Perífrasis verbal modal: “debe permitir”. Verbo nominalizado: “el registro”. Complemento: “de los clientes”
Es importante que la secretaria obtenga los teléfonos.	FMOB2	Complemento A: “es importante”. Que: “que”. Sintagma nominal: “la secretaria” Verbo: “obtenga”. Complemento: “los teléfonos”
Garantizar que el director guarde la información de los clientes.	FMOB3	Verbo en infinitivo: “garantizar”. Que: “que”. Sintagma nominal: “el director”. Verbo: “guarde” Complemento: “ la información de los clientes”
El sistema debe validar los datos	FMOB4	Sintagma Nominal: “El sistema”; Perífrasis verbal modal: “debe validar”; Complemento: “los datos”.

Fuente: autores

De acuerdo con los datos registrados en la Tabla I, las reglas morfosintácticas para la identificación de objetivos quedan determinadas de la siguiente manera:

$$Si \ FMOB1(S) \ FMOB2(S) \ FMOB3(S) \ FMOB4(S) \\ \rightarrow \ Objetivo(nombre:\{inf(S_{vp}) + S_{vp+}\})$$

Esta regla indica que si una oración sí cumple con alguna de las formas morfosintácticas de los objetivos, la salida es un objetivo. El parámetro que se pasa es el nombre que se le da al objetivo, siendo, para esta regla, el verbo principal en infinitivo ($inf(s_{vp})$, más el conjunto de palabras que se encuentren después del verbo principal (vp).

2.2 Reglas morfosintácticas para identificar agentes

Como los agentes son componentes activos del sistema que juegan un rol específico en la satisfacción de un objetivo [14], se define la siguiente estructura gramatical:

$$FMAG1: \ Sustantivo \ o \ sintagma \ nominal + Verbo \ no \ copulativo + Complemento$$

Es importante que el verbo sea no copulativo, porque de esta forma se garantiza que la oración no es nominativa, sino que el sujeto realiza una acción. En la Tabla II se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar agentes.

De acuerdo con lo presentado en la Tabla II, las reglas morfosintácticas para la identificación

de agentes quedan determinadas de la siguiente manera:

$$Si \ FMAG1(S) \rightarrow agente(nombre:SN)$$

Es decir, si una oración presenta la estructura morfosintáctica definida para los agentes, se crea una instancia del mismo, pasándole como parámetro el nombre del agente que es, para esta regla, el sintagma nominal (SN).

2.3 Reglas morfosintácticas para identificar entidades y atributos

Una entidad es un objeto pasivo y autónomo, que puede tener uno o varios atributos que lo describen [14]. En este sentido, se definen las siguientes reglas para su identificación:

FME1: si en una frase se encuentran dos sustantivos unidos por la preposición “de” o “del” el segundo sustantivo será una entidad, y el primer sustantivo será candidato a ser atributo de la entidad identificada. *Sustantivo + de/del + sustantivo.*

FME2: si en una frase se encuentran dos sustantivos unidos por el verbo “tiene” el primer sustantivo será una entidad, y el segundo sustantivo será candidato a ser atributo de la entidad identificada. *Sustantivo + tiene + determinante (opcional) + sustantivo.*

FME3: si un sustantivo o sintagma nominal es identificado como entidad y como posible atributo tiene prelación la entidad.

En la Tabla III, se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar entidades y atributos.

TABLA II
EJEMPLO DE LAS FORMAS MORFOSINTÁCTICAS DEFINIDAS PARA CARACTERIZAR OBJETIVOS

Lenguaje natural	Elementos morfosintácticos identificados	Agente identificado
El personal de ambulancia envía información	Sintagma nominal: “El personal de ambulancia”. Verbo principal: “envía” (no copulativo). Complemento: “información”	“personal de ambulancia”
El sistema CAD puede proponer otra ambulancia	Sintagma nominal: “El sistema CAD” Verbo principal: “puede” (no copulativo) Complemento: “proponer otra ambulancia”	“sistema CAD”
El controlador es de la empresa CX	Sintagma nominal: “El controlador” Verbo principal: “es” (copulativo) Complemento: “proponer otra ambulancia”	No hay agente (el verbo principal es copulativo)

Fuente: autores

TABLA III
EJEMPLO DE FORMA MORFOSINTÁCTICA PARA CARACTERIZAR ENTIDADES Y ATRIBUTOS

Lenguaje natural	Forma utilizada	Entidad	Atributo
El radio operador recibe información del personal de la ambulancia.	FME1	Persona	Información
Cada ambulancia tiene una placa.	FME2	Ambulancia	Placa

Fuente: autores

De acuerdo con lo presentado en la Tabla III, las reglas morfosintácticas para la identificación de entidades quedan determinadas de la siguiente manera:

$$Si \ FME1(S) \rightarrow entidad(nombre:S2) \\ \wedge atributo(nombre:S1)$$

Así mismo, si una oración cumple con la forma FME1, se crea una instancia de la entidad y un atributo. El parámetro que se pasa es el nombre que se le da a la entidad que, en este caso, es el segundo sustantivo del sintagma nominal (S2). También se crea un atributo cuyo parámetro es el nombre que corresponde al primer sustantivo del sintagma nominal (S1).

$$Si \ FME2(S) \rightarrow entidad(nombre:S1) \\ \wedge atributo(nombre:S2)$$

$$\forall (entidad(nombre:S1) \wedge atributo(nombre:S2)): \\ S2 = S1 \rightarrow entidad(nombre:S1)$$

Así, si un sustantivo o sintagma nominal es identificado como entidad y como posible atributo tiene prelación la entidad.

2.4 Reglas morfosintácticas para identificar operaciones

Una operación es una relación binaria entre estados del sistema. Posee una tupla de variables de entrada, que designan una instancia de un objeto cuyo estado determina la aplicación de la

operación; también tiene una tupla de variables de salida, la cual indica una instancia de un objeto cuyo estado cambia con la aplicación de la operación. De esta forma, se definen las siguientes estructuras morfosintácticas a fin de identificar estos elementos:

FMOP1. Si: {verboprincipal $vp \in$ lista verbos de operación} Y {objeto directo $dobj +$ "de" + objeto preposicional $obj\text{-}prep$ } \rightarrow operación (nombre: $vp + lema(dobj)$), input ($dobj, obj\text{-}prep$), output($dobj$, SUSTANTIVAR(vp)).

La forma FMOP1 establece que, dada una operación de la forma indicada, se identifica la variable de entrada que consta del objeto directo y el preposicional, y la variable de salida que se compone del objeto directo y la sustantivación del verbo principal.

FMOP2. Si: {verboprincipal $vp \in$ lista verbos de operación} Y { $(\exists dobj)(\nexists obj\text{-}prep) \rightarrow$ operación (nombre: $vp + lema(dobj)$), input($dobj$), output($dobj$, SUSTANTIVAR(vp)).

La forma FMOP2 indica que, si una frase con un verbo principal catalogado como de operación, posee en su complemento un objeto directo, pero no preposicional, la variable de entrada será este objeto, y la de salida, el mismo objeto pero acompañado de la sustantivación del verbo principal.

En la Tabla IV se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar operaciones.

TABLA IV
EJEMPLO DE FORMA MORFOSINTÁCTICA PARA CARACTERIZAR OPERACIONES

Lenguaje natural	Forma utilizada	Verbo de operación	Objeto directo	Operación	Input	Output
El operario consulta la localización de la ambulancia.	FMOP1	Consultar	Localización	Consultar localización	Localización de la ambulancia	Consulta de la localización
El sistema registra a los usuarios	FMOP2	Registrar	Usuario	Registrar usuario	Usuario	Registro de usuario

Fuente: autores

3. REGLAS SEMÁNTICAS PARA LA IDENTIFICACIÓN DE LOS ELEMENTOS BÁSICOS DEL DIAGRAMA DE OBJETIVOS DE KAOS

A continuación, se plantean cuatro reglas con el objeto de clasificar los objetivos, agentes y operaciones y caracterizar las entidades. Para lograrlo, se desambiguan los conceptos principales de cada elemento.

Existen múltiples métodos para realizar desambiguaciones semánticas, los cuales tienen como propósito principal determinar el mejor sentido (o synset) posible para una palabra en una oración dada. Entre ellos, se utiliza el Lesk simplificado [16], el cual toma las definiciones de todos los sentidos de la palabra por desambiguar, y lo compara con las palabras que están en el contexto, es decir, en su vecindad, eligiendo aquel sentido cuya definición tenga más palabras en común con la vecindad. Sin embargo, en las bases de datos léxicas en español, existe una gran carencia de definiciones para los sentidos, razón por la cual este método no arroja ningún resultado en muchos casos. Por esto, se plantea como segunda opción utilizar el método del Sentido Más Frecuente -MFS- [19] o el UKB [3], según se indique en el momento de la ejecución del sistema.

3.1 Clasificación de objetivos

Una vez que se tiene la caracterización morfosintáctica de los objetivos (presentada en la sección anterior), se realiza el proceso de desambiguación del verbo principal del objetivo (V_m). Su estructura es la siguiente: $V_m \rightarrow ILI-XXXXXX-V$, donde XXXXXX representa el identificador universal del synset.

El sentido seleccionado a través de los métodos de desambiguación utilizados permitirá obtener un identificador único para el sentido del verbo y una definición (gloss) asociada al mismo. Una vez obtenido el sentido del verbo, se procede a realizar su clasificación, lo que permitirá clasificar al objetivo. Un objetivo se clasifica según su tipo en: mantenimiento, alcanzar, evitar y parar [14].

Se considera "M" como el conjunto de sentidos definidos previamente de los verbos de mantenimiento, donde cada sentido estará en el conjunto $M = \{M1, \dots, Mn\}$. Análogamente esta misma especificación seguirá para los demás grupos de

sentidos de verbos: Logro: $L = \{L1, \dots, Ln\}$; Parar: $P = \{P1, \dots, Pn\}$; Evitar: $E = \{E1, \dots, En\}$.

Dado lo anterior, la clasificación de los objetivos en cualquiera de los tipos será determinado por el siguiente conjunto de ecuaciones (1) - (4).

$$\varphi_M = \frac{\sum_{i=1}^n |Similitud(M_i, V_m)|}{n} \quad (1)$$

$$\varphi_L = \frac{\sum_{i=1}^n |Similitud(L_i, V_m)|}{n} \quad (2)$$

$$\varphi_P = \frac{\sum_{i=1}^n |Similitud(P_i, V_m)|}{n} \quad (3)$$

$$\varphi_E = \frac{\sum_{i=1}^n |Similitud(E_i, V_m)|}{n} \quad (4)$$

Donde similitud es la medida de la interrelación existente entre dos palabras cualquiera en un texto, dada por la (5) propuesta por [23]:

$$similitud(concept_1, concept_2) = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \quad (5)$$

Donde depth (LCS) es la distancia completa del árbol de taxonomía usado (generalmente Wordnet, que es una base de datos léxica de relaciones semánticas entre términos, tales como hiperónimos, hipónimos y sinónimos) y depth (concept) representa la profundidad del concepto en el árbol de taxonomía usado.

Una vez se tiene la medida del promedio de similaridad (φ) de cada grupo de objetivos, se realiza el comparativo entre los coeficientes y el mayor coeficiente permitirá realizar la clasificación del verbo en el grupo correspondiente de acuerdo con los tipos objetivos definidos en la metodología KAOS.

3.2 Clasificación de agentes

Un agente puede ser clasificado como agente de *software* o de ambiente. Considerada $A = \{A1, A2, A3, A4, \dots, An\}$ como una lista de posibles agentes identificados por medio de reglas morfosintácticas, se realiza el proceso de desambiguación de los sustantivos del agente ($Sp1, Sp2, \dots, Spm$), por tanto, un agente queda definido de la siguiente manera: $A_i = \{Sp1, Sp2, \dots, Spm\}$, donde:

Sp1 → ILI-XXXXXX-N; Sp2 → ILI-XXXXXX-N; Spm → ILI-XXXXXX-N; XXXXXX representa el identificador universal del synset.

Una vez se tiene identificado el sentido correcto para cada sustantivo (articulado a cada agente), se aplica el siguiente procedimiento:

Un agente se considera de *software* si y solo si:

$$\exists S_{pi} \in A_j: \max hyp(S_{pi}) = "Machine"$$

Un agente se considera de ambiente si y solo si:

$$\exists S_{pi} \in A_j: \max hyp(S_{pi}) = "Human"$$

Donde maxhyp (Spi) corresponde al sentido del último hiperónimo en el árbol de taxonomías del MCR o de Wordnet.

3.3 Clasificación de operaciones

Se toma el conjunto de posibles operaciones $O = \{O1, O2, \dots, On\}$ que cumplen el proceso de identificación morfosintáctico. Luego, para la identificación semántica de operaciones se realiza el proceso de desambiguación del verbo principal de la operación (svp). El sentido correcto del verbo se obtiene mediante los métodos propuestos en este artículo. Por tanto, una operación queda definida de la siguiente manera: $Svp \rightarrow ILI-XXXXXX-V$, donde XXXXXX representa el identificador universal del synset.

Adicionalmente, se deberá cumplir la siguiente regla: sea $P = \{P1, P2, \dots, Pn\}$ el conjunto de sentidos de los verbos de operación, clasificados por Jaramillo, Zapata y Arango (2005). De esta forma, una operación será clasificada si y solo si (6):

$$\mu > \frac{\sum_{i=1}^n |Similitud(P_i, S_{vp})|}{n} \quad (6)$$

Donde μ corresponde a un umbral definido por el usuario con el fin de determinar qué tan estricta es la condición de ser operación o no, y similitud hace referencia a la ecuación (5).

3.4 Caracterización de entidades

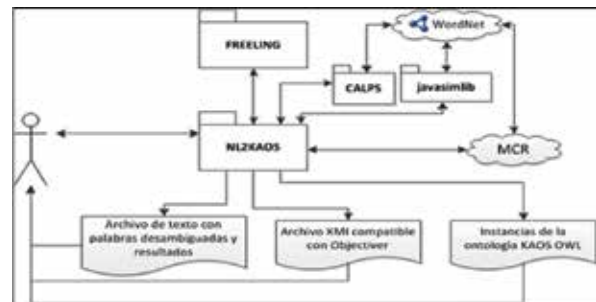
Una vez que se tiene la caracterización morfosintáctica de las entidades, se realiza el proceso de desambiguación del sustantivo que compone la entidad (Em). Para encontrar el sentido correcto del sustantivo se utilizaron los métodos de desambiguación citados en este artículo. Dado lo

anterior, queda asignado el sentido al sustantivo principal de la entidad. Em → ILI-XXXXXX-N, donde XXXXXX representa el identificador universal del synset. Adicionalmente, se toma la definición de ese sentido, para asociarlo como la definición del elemento.

4. AUTOMATIZACIÓN DEL MÉTODO PROPUESTO

La arquitectura general del *software* "NL2KAOS" desarrollado para la validación del método propuesto se presenta en la Fig. 3. Este sistema permite: (i) minimizar la ambigüedad semántica presente en el universo del discurso, (ii) obtener automáticamente los elementos básicos del diagrama de objetivos KAOS, (iii) la participación directa del interesado durante la identificación y validación de los requisitos de *software*, (iv) la desambiguación semántica de cada palabra, y (v) la instanciación de los elementos identificados a OWL de la ontología KAOS desarrollada.

Fig. 3. ARQUITECTURA GENERAL DEL MÉTODO PROPUESTO



Fuente: autores.

La entrada del sistema es el texto en español, sin errores ortográficos, y haciendo uso de la voz activa para explicitar a los sujetos de las acciones. El sistema NL2KAOS incorpora el conjunto de reglas descritas anteriormente, las cuales se aplican una vez se ha realizado el tokenizado, lematizado, etiquetador morfosintáctico y análisis de dependencias, por parte de la librería Freeling 3.0 [20]. Para aplicar las reglas semánticas se consulta el WordNet en español, el cual está disponible mediante el Multilingual Central Repository (MCR) [7]. Para calcular la similitud semántica se utilizan las librerías CALPS [9] y javasimlib [22]. Finalmente, caracterizados todos los elementos identificados se exportan los resultados en instancias de la ontología KAOS en OWL desarrollada, un archivo

XMI compatible con Objectiver (*software* utilizado en la diagramación de KAOS) y la lista de palabras desambiguadas.

5. RESULTADOS Y DISCUSIÓN

Para validar el método propuesto, en esta sección se registran los resultados de experimentación realizados. Estos se enfocan en valorar los aspectos de completitud, correctitud y consistencia [26].

Completitud [23], establecen varias métricas para medir la efectividad de modelos de recuperación de información. En el caso del procesamiento de lenguaje natural, también existen estas métricas. Concretamente, en el proceso de identificación y clasificación de elementos KAOS, se puede utilizar el alcance (recall), dado que permite determinar la fracción de elementos realmente presentes (identificados por un analista) y que han sido hallados por el método propuesto. El alcance se halla para cada elemento (objetivo, entidad, operación y agentes) de acuerdo con la expresión (7)

$$\text{Alcance}_{\text{elemento}} = \frac{|\text{Elementos hallados} \cap \text{Elementos presentes}|}{|\text{Elementos presentes}|} \times 100 \quad (7)$$

Este índice se refiere a la completitud, debido a que evalúa la proporción entre aquellos elementos que el sistema halla y los que realmente están presentes.

Correctitud. Otro aspecto por tener en cuenta es la fracción de elementos que el sistema halló frente al conjunto de elementos realmente presentes. Este índice se conoce como precisión, y también debe realizarse ponderando los resultados de cada uno de los elementos básicos del diagrama de objetivos de KAOS. La precisión (8) indica qué tanto de los elementos hallados y presentes están en la cantidad de elementos hallados por el sistema propuesto.

$$\text{Precisión}_{\text{elemento}} = \frac{|\text{Elementos hallados} \cap \text{Elementos presentes}|}{|\text{Elementos hallados}|} \times 100 \quad (8)$$

Consistencia. La consistencia permite verificar que los elementos obtenidos estén acordes con la información dada por el interesado, es decir, que no se cambie el sentido y necesidad del mismo. De manera que este aspecto se puede evaluar con la misma correctitud, ya que midiendo la precisión se establece la proporción entre los elementos hallados y verdaderamente presentes.

El método se probó en diez casos de estudio, tres de ellos han sido descritos previamente en la literatura científica: (i) el sistema de servicio de ambulancias de Londres [17], (ii) el sistema de una pizzería [25], y (iii) sistema asociado a un elevador [21]. Es importante resaltar que estos casos de estudio fueron adaptados al método propuesto en este artículo. En las Tablas V y VI se presenta un compendio de los resultados obtenidos para los diez casos de estudio.

TABLA V
RESUMEN DE RESULTADOS PARA OBJETIVOS, AGENTES Y ENTIDADES EN LOS CASOS DE ESTUDIO

CASO DE ESTUDIO	Objetivos		Agentes		Entidades	
	Alcance	Precisión	Alcance	Precisión	Alcance	Precisión
Ambulancia	100,00%	88,89%	100,00%	85,71%	78,95%	60,00%
Ascensor	92,86%	86,67%	100,00%	57,14%	56,00%	48,28%
Pizzería	93,75%	93,75%	100,00%	62,50%	64,29%	69,23%
Carrefour	100,00%	100,00%	100,00%	66,67%	86,67%	76,47%
Colácteos	100,00%	87,50%	100,00%	75,00%	100,00%	71,43%
Colcerámicas	100,00%	100,00%	100,00%	87,50%	100,00%	68,00%
ENCA	100,00%	80,00%	100,00%	85,71%	100,00%	80,95%
Guayacán	81,82%	69,23%	88,89%	80,00%	100,00%	58,82%
Marinozzi	100,00%	85,71%	100,00%	83,33%	85,71%	75,00%
Yamaha	100,00%	100,00%	100,00%	85,71%	100,00%	82,35%
Promedio	96,84%	89,18%	98,89%	76,93%	87,16%	69,05%
Desviación estándar	0,057	0,094	0,033	0,105	0,155	0,102

Fuente: autores

Es importante resaltar que las librerías (CALPS y javasimlib) utilizadas para calcular la similitud semántica y los algoritmos de desambiguación, no afectan la identificación de los objetivos, entidades y agentes, es decir, las reglas semánticas están definidas solo para cumplir con la clasificación de los elementos. Sin embargo, para el caso de las operaciones, se debe corroborar si estas tienen relación semántica con la lista de los verbos de operación. Dado lo anterior, las operaciones pueden cambiar de acuerdo con la librería de similitud semántica y la técnica de desambiguación.

Con base en los resultados obtenidos a través de las pruebas realizadas, se puede indicar que: (i) los objetivos se logran identificar con un alto alcance y precisión (96,84% y 89,18%); (ii) los agentes tienen un muy buen alcance (98,89%), aunque disminuye su calidad en la precisión

(76,93%); (iii) con respecto a las entidades, se tienen índices inferiores a los objetivos y entidades, pero un alcance de 87,16% es aceptable, aunque la precisión disminuye a un 69,05%; (iv) las operaciones arrojan un alcance promedio de 71,86% y precisión promedio de 75,85%.

Las debilidades en la identificación de las entidades obedecen a la dificultad de identificar y asignar los atributos a la entidad respectiva. Para el caso de las operaciones, es posible que la lista de operaciones no sea lo suficientemente completa, ya que hubo operaciones que se identificaron morfosintácticamente, pero en la evaluación de la relación semántica con dicha lista, muchos candidatos eran eliminados, aunque en realidad eran operaciones. Esto evidencia incluso la debilidad de la librería javasimlib para medir la similitud semántica, ya que presenta los resultados más bajos.

TABLA VI
RESULTADOS PARA LAS OPERACIONES EN LOS CASOS DE ESTUDIO

Prueba	PARÁMETRO	MFS		UKB	
		CALPS	javasimlib	CALPS	javasimlib
AMBULANCIA	Alcance	71,43%	6,67%	80,00%	26,67%
	Precisión	100,00%	100,00%	85,71%	100,00%
ASCENSOR	Alcance	66,67%	11,11%	66,67%	33,33%
	Precisión	85,71%	100,00%	85,71%	100,00%
PIZZERÍA	Alcance	50,00%	12,50%	50,00%	12,50%
	Precisión	100,00%	100,00%	100,00%	100,00%
Carrefour	Alcance	100,00%	50,00%	100,00%	100,00%
	Precisión	50,00%	50,00%	50,00%	66,67%
Colácteos	Alcance	100,00%	50,00%	100,00%	50,00%
	Precisión	50,00%	100,00%	50,00%	50,00%
Colcerámicas	Alcance	100,00%	100,00%	100,00%	100,00%
	Precisión	33,33%	50,00%	33,33%	66,67%
ENCA	Alcance	100,00%	50,00%	100,00%	100,00%
	Precisión	40,00%	100,00%	40,00%	66,67%
Guayacán	Alcance	100,00%	40,00%	100,00%	80,00%
	Precisión	83,33%	100,00%	83,33%	100,00%
Marinozzi	Alcance	66,67%	66,67%	66,67%	66,67%
	Precisión	40,00%	66,67%	40,00%	66,67%
Yamaha	Alcance	100,00%	100,00%	100,00%	100,00%
	Precisión	100,00%	100,00%	100,00%	100,00%

Fuente: autores

Finalmente, al obtener un promedio general de los resultados arrojados por sistema NL2KAOS, para los experimentos realizados, se logra establecer un alcance del 88,69% y una precisión del 77,75%. En consecuencia, se puede indicar que los resultados obtenidos son buenos, sin embargo, quedan aspectos por mejorar con relación al alcance y la precisión. Los resultados obtenidos demuestran que es posible realizar un buen trabajo de identificación y clasificación de los elementos básicos del diagrama de objetivos de KAOS, utilizando técnicas de procesamiento de lenguaje natural. En términos generales, se puede indicar que, el método propuesto en este artículo logra identificar satisfactoriamente los agentes y los objetivos. Sin embargo, las entidades y, especialmente, las operaciones, constituyen la mayor falencia del método.

CONCLUSIONES

Con el propósito de presentar algunas soluciones a la problemática relacionada con la educación y validación de requisitos de *software*, a través del diagrama de objetivos de KAOS, se propuso en este artículo crear un método de procesamiento terminológico que permitiera: (i) caracterizar los elementos básicos a partir del uso del lenguaje natural en idioma español, (ii) minimizar la ambigüedad semántica presente en el universo del discurso, (iii) obtener automáticamente los elementos básicos, y (iv) elaborar automáticamente dicho diagrama. Este método incluyó:

Desarrollar una ontología con base al metamodelo del diagrama de objetivos de KAOS propuesto por Matulevičius [18]. La creación de esta Ontología permitió una mejor definición y caracterización de los elementos que conforman el diagrama de objetivos de KAOS.

Definir cuatro formas morfosintácticas para caracterizar objetivos, una para caracterizar agente, tres para caracterizar entidades y dos para caracterizar operaciones. La definición de las formas morfosintácticas tiene como propósito principal brindar una solución a la necesidad de identificar la trazabilidad que debe existir entre el lenguaje natural y los elementos básicos (objetivo, entidad, agente y operación) del diagrama de objetivos de KAOS.

Definir cuatro reglas semánticas. Una vez definidas las formas morfosintácticas se procedió con la definición de las reglas semánticas, las cuales permiten mejorar procesamiento terminológico propuesto, puesto que, permiten minimizar la ambigüedad semántica. Para lo que se aplicaron los métodos Lesk simplificado, MFS y UKB.

La validación del método se realizó mediante casos de prueba registrados en la literatura científica, además, se tuvieron en cuenta las características completitud, correctitud y consistencia [26]. Esta validación permitió concluir que: (i) el método propuesto y validado a través del sistema NL2KAOS, posee en promedio, un alcance del 88,69% y una precisión del 77,75%, (ii) la automatización del diagrama de objetivos de KAOS (utilizado para la identificación y validación de requisitos de *software*), elaborado en esta propuesta, reduce significativamente el tiempo utilizado frente al mismo proceso realizado de forma manual por parte del analista, (iii) mejora la trazabilidad que debe existir entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos de KAOS, (iv) minimiza la ambigüedad semántica de tipo polisémica presente en el universo del discurso, (v) incluye la validación semiautomática de requisitos de *software* por parte del interesado en la etapa inicial del ciclo de vida del *software*, y (vi) mejora las características de correctitud, completitud y consistencia durante el proceso de identificación y validación de requisitos mediante el uso del diagrama de objetivos de KAOS.

AGRADECIMIENTOS

Este artículo se realizó en el marco del proyecto de investigación: “Modelo de procesamiento terminológico basado en ontologías para la desambiguación verbal en la educación de requisitos de *software*”. Hermes código: 18717 Universidad Nacional de Colombia at Medellín).

REFERENCIAS

- [1] D Alan, “Software Requirements, Objetos, Funciones y Estados,” Editorial Prentice Hall, New Jersey, USA, 1993.
- [2] F. Almisned, J. Keppens, “Requirements Analysis: Evaluating KAOS Models. Software Engineering & Applications,” Vol. 3, 2010, pp. 869-874.

- [3] E. Aguirre, y A. Soroa, "Personalizing pagerank for word sense disambiguation," *Actas de la 12a Conferencia del capítulo europeo de la Association for Computational Linguistics*, Stroudsburg, USA, 30 de marzo al 3 de abril, 2009, pp. 33-41.
- [4] V. Demonte, y I. Bosque, "Gramática descriptiva de la lengua española," Editorial Espasa, Madrid, España, 1999.
- [5] DRAE: Diccionario de la Real Academia de la lengua española (en línea), <http://www.rae.es/rae.html>. Acceso: 2 de mayo, 2014.
- [6] F. Genta, "Perífrasis Verbales en español: Focalización aspectual, restricción temporal y rendimiento discursivo," Tesis Doctoral. Universidad de Granada, España, 2008.
- [7] A. González, E. Laparra, y G. Rigau, "Multilingual Central Repository version 3.0," *Actas de la 8a conferencia internacional en Recursos y Evaluación del Lenguaje (LREC'12)*, pp. 2525-2529, Estambul, Turquía, 2012. Disponible en: <http://adimen.si.ehu.es/web/MCR/>. Acceso: 2 de mayo 2014.
- [8] J. Guzmán-Luna, L. Lezcano y S. Gómez, "Obtaining Agents and Entities from Natural Language," *Lecture Notes in Electrical Engineering*, vol. 312, pp. 29-36, 2015.
- [9] D. Hope, "Java WordNet::Similarity, Cognitive and Language Processing Systems Group (CALPS)," University of Sussex (en línea), 2008. <http://www.sussex.ac.uk/Users/drh21/>. Acceso: 2 de mayo 2014.
- [10] J. Ivar, B. Grady, y R. James, "El Proceso Unificado de Desarrollo de Software," Pearson Addison-Wesley, 2001.
- [11] A. Jaramillo, C. Zapata, y F. Arango, "Una propuesta para el reconocimiento semiautomático de operaciones utilizando un enfoque lingüístico," *Revista Facultad de Ingeniería*, vol. 34, 42-51 (2005).
- [12] A. Lamsweerde, G. Dardenne, y S. Fichas, "Goal-Directed Requirements Acquisition," *Science of Computer Programming*, vol. 20, no. 2, pp. 3-50, 1993.
- [13] A. Lamsweerde, "Requirements Engineering in the Year 2000: A Research Perspective," *Actas de la 22a Conferencia Internacional en Ingeniería del Software*, Limerick, Irlanda, 7 a 9 de junio, 2000, pp. 449-458.
- [14] Lamseerde, A., "Requirements Engineering: From system goals to UML models to software specifications", 1a edición. John Wiley and Sons, West Sussex, Inglaterra, 2009.
- [15] A. Lamsweerde, y E. Letier, "From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. Radical Innovations of Software and Systems Engineering in the Future," Springer, 2004, pp. 153-1664.
- [16] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," *Actas de la 5a Conferencia Internacional Annual en Documentación de Sistemas*, New York, NY, USA, 1986, pp. 24-26.
- [17] E. Letier, "Reasoning about Agents in Goal-Oriented Requirements Engineering". Tesis de Doctorado, Département d'Ingénierie Informatique, Université catholique de Louvain, Bélgica, 2001.
- [18] R. Matulevičius, P. Heymans y A. Opdahl, "Ontological Analysis of KAOS Using Separation of Reference". En: *IGIPublishing*, Vol. 6, 2007, pp. 37-54.
- [19] H. Ng, y H. Lee, "Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach," *Actas de la 34a Conferencia Anual de la Association for Computational Linguistics*, pp. 40-47. Stroudsburg, USA. 1996.
- [20] Padró, L. y Stanilovsky, E., "Freeling 3.0: towards wider multilinguality," *Actas de la 8a Conferencia Internacional en Recursos y Evaluación del Lenguaje (LREC 2012)*, Estambul, Turquía, 2012, pp. 2473-2479.
- [21] Respect IT. "A KAOS Tutorial. Objectiver," (en línea), 2007. <http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>. Acceso: 2 de mayo, 2014.
- [22] N. Seco, T. Veale, y T. Hayes, "An Intrinsic Information Content Metric for Semantic Similarity in WordNet," *Actas de la 16a Conferencia Europea en Inteligencia Artificial*, Valencia, España, 2004, pp. 1089-1090.
- [23] Z. Wu y M. Palmer, "Verb semantics and lexical selection," *Actas de la 32a Conferencia Anual de la Association for Computational Linguistics*, Las Cruces, New México, 1994.
- [24] C. M. Zapata, F. Vargas, "Innovación en el diseño y evaluación de proyectos: establecimiento de las relaciones lingüísticas entre objetivos y problemas," *Lámpskos*, vol. 3, no. 6, pp. 46-55, 2011.
- [25] C. M. Zapata, S. Villegas, y F. Arango, "Reglas de consistencia entre modelos de requisitos de UN-Método," *Revista Universidad Eafit*, vol 141, no. 42, pp. 40-59, 2006.
- [26] D. Zowghi, y V. Gervasi, "The Three Cs of requirements: consistency, completeness, and correctness," *Actas del 8 | taller internacional en Ingeniería de Requisitos: Fundamentos para la Calidad del Software*, Essen, Germany, 9 al 10 de septiembre, 2002, pp. 155-164.