

VALIDACIÓN DEL MÉTODO PARA LA OBTENCIÓN AUTOMÁTICA DEL DIAGRAMA DE OBJETIVOS DESDE ESQUEMAS PRECONCEPTUALES

CARLOS MARIO ZAPATA*
LUIS ALFONSO LEZCANO**
PAULA ANDREA TAMAYO***

RESUMEN

Según el CDM (*Custom Development Method*), el desarrollo de aplicaciones de *software* suele empezar con una fase de definición, en la cual se determinan los procesos que se realizan en la organización que requiere el *software*, los problemas que motivan tal desarrollo y, especialmente, los objetivos asociados con las diferentes áreas de la organización. En esta fase, el diagrama de objetivos de KAOS (*Knowledge Acquisition Automated Specification*) se suele utilizar para describir los objetivos de alto nivel de la organización y dividirlos paulatinamente en subobjetivos hasta alcanzar los requisitos y expectativas de los interesados. El grupo de Ingeniería de *Software* de la Escuela de Sistemas de la Universidad Nacional de Colombia desarrolló un método para automatizar la obtención del diagrama de objetivos de KAOS a partir de esquemas preconceptuales, que son diagramas que describen los procesos y el vocabulario de una organización que desee desarrollar una aplicación de *software*. En este artículo se realiza la validación de dicho método, empleando para ello tres casos de estudio incluidos en la literatura especializada.

PALABRAS CLAVE: diagrama de objetivos de KAOS; esquema preconceptual; objetivo; validación.

* Ingeniero Civil, Especialista en Gerencia de Sistemas Informáticos, Magíster en Ingeniería de Sistemas y Doctor en Ingeniería con énfasis en Sistemas. Profesor Asociado de la Escuela de Sistemas, Universidad Nacional de Colombia, Sede Medellín. Grupo de Investigación en Ingeniería de Software. cmzapata@unal.edu.co

** Ingeniero de Sistemas, Estudiante de Maestría en Ingeniería de Sistemas de la Universidad Nacional de Colombia. Grupo de Investigación en Ingeniería de Software de la Escuela de Sistemas, Universidad Nacional de Colombia, sede Medellín. lalezcan@unal.edu.co

*** Ingeniera de Sistemas e Informática y Magíster en Ingeniería de Sistemas de la Universidad Nacional de Colombia. Grupo de Investigación en Ingeniería de Software de la Escuela de Sistemas, Universidad Nacional de Colombia, sede Medellín. patamayo@unalmed.edu.co

ABSTRACT

According to the Custom Development Method (CDM), the first phase of the software development process is commonly the definition phase. Processes related to the organization in which the software application is needed, problems that motivates the development process, and objectives associated with several organizational areas are determined in this phase. KAOS (Knowledge Acquisition Automated Specification) goal diagram is used in this phase to describe high-level organizational goals and then divide them into sub-objectives concerned with the stakeholder needs and expectations. Software Engineering group of the Escuela de Sistemas of the Universidad Nacional de Colombia developed a method to automate the KAOS goal diagram obtaining from Pre-conceptual Schemas, which are diagrams to describe the organizational processes and vocabulary linked with the software development. We use in this paper three case studies in order to validate such a method. The case studies are reported in specialized papers about this issue.

KEY WORDS: KAOS goal diagram; pre-conceptual schema; goal; validation.

1. INTRODUCCIÓN

La compañía Oracle (2000), en su método de desarrollo de *software* CDM, estableció como etapa inicial del desarrollo de una aplicación de *software* la “definición”, que incluye un proceso llamado educación de requisitos, en el cual las necesidades y expectativas de los interesados –aquéllos con algún tipo de interés en que el *software* se realice– se recolectan para traducirlas finalmente en especificaciones formales y semiformales. Los analistas se encargan de este proceso y lo realizan mediante reuniones con los interesados (es de notar que el término “interesado” equivale al vocablo inglés *stakeholder* y se refiere a todas las personas que poseen algún tipo de interés en el desarrollo de una aplicación), revisión de documentación y otras técnicas de educación que buscan descubrir los procesos de la organización a la cual se le va a construir la aplicación, los problemas que motivan el surgimiento de la aplicación y los objetivos de la organización y de la aplicación de *software* que se va a construir. Lamsweerde *et al.* (1993) propusieron el diagrama de objetivos de KAOS, con el fin de representar jerárquicamente estos objetivos, de forma tal que se colocaran los de alto nivel de la organización en la raíz del árbol

y los de bajo nivel (más operativos y relacionados con los requisitos y expectativas del *software*) en la parte inferior. Además, en este diagrama se pueden asignar responsabilidades a los diferentes actores de la organización.

La construcción del diagrama de objetivos de KAOS es un proceso manual en el que los analistas deben interpretar el discurso del interesado y extraer de allí los principales objetivos de la organización, así como los requisitos y expectativas de los interesados en relación con el *software*. Como consecuencia de lo anterior, se obtienen diagramas que no contienen todos los elementos que se pueden identificar, se utiliza en ocasiones una sintaxis diferente a la definida por Lamsweerde *et al.* (1993) o se generan ciertas inconsistencias con el modelo verbal presentado. Debido a esto, el grupo de Ingeniería de *Software* de la Universidad Nacional de Colombia, Sede Medellín, propuso un método para el trazado automático de este diagrama (Lezcano, 2007), tomando como punto de partida los denominados “esquemas preconceptuales” (Zapata *et al.*, 2006a, 2006b y 2006c).

En este artículo, se realiza la validación de dicho método, empleando para ello tres casos de



estudio presentados por Letier (2001), Heaven y Finkelstein (2004) y Zapata *et al.* (2006d). Esta validación se lleva a cabo mediante la comparación y el análisis de las características de completitud, consistencia y corrección entre los diagramas obtenidos con la aplicación del método del grupo de Ingeniería de Software y los presentados en los casos de estudio señalados.

La estructura del resto del artículo es la siguiente: en la sección 2 se presenta el marco teórico que fundamenta esta validación; en la sección 3 se muestran los principales casos de estudio en la elaboración del diagrama de objetivos y se discute la validez de los resultados en cada caso; en las secciones 4 y 5 se presentan las conclusiones y el trabajo futuro que se pueden derivar de este artículo.

2. MARCO TEÓRICO

2.1 Diagrama de objetivos de KAOS

Según Lamsweerde *et al.* (1993), el diagrama de objetivos de KAOS permite al analista descubrir, mediante la identificación de los objetivos generales de la organización, los objetivos específicos que justifican la construcción del *software*. El proceso para el trazado del diagrama de objetivos de KAOS requiere que se definan los objetivos secundarios que subrogan los objetivos generales, para luego presentar los objetivos más elementales que los subrogan, y así sucesivamente, hasta que se alcancen

expectativas y requisitos del interesado en relación con la aplicación de *software* que se piensa construir. En la figura 1 se pueden observar los principales elementos de este diagrama.

La explicación de los símbolos es la siguiente:

- Objetivo. Fin que se espera lograr con un proceso o actividad o incluso con la misión de una organización.
- Requisito. Un objetivo en el nivel de la solución informática que no será negociable dentro del proceso de desarrollo.
- Expectativa. Un objetivo en el nivel de la solución informática que no se espera que se incorpore a ella.
- Actor. Responsable de un requisito o una expectativa.
- Propiedad del dominio. Una característica que se requiere para que los objetivos se alcancen a un determinado nivel.
- Conectores. Flechas que vinculan objetivos, expectativas, requisitos, propiedades del dominio y actores.

2.2 Esquemas preconceptuales

Según Zapata *et al.* (2006a, 2006b y 2006c), los esquemas preconceptuales son diagramas que permiten la representación de la terminología de un dominio para facilitar su traducción posterior a

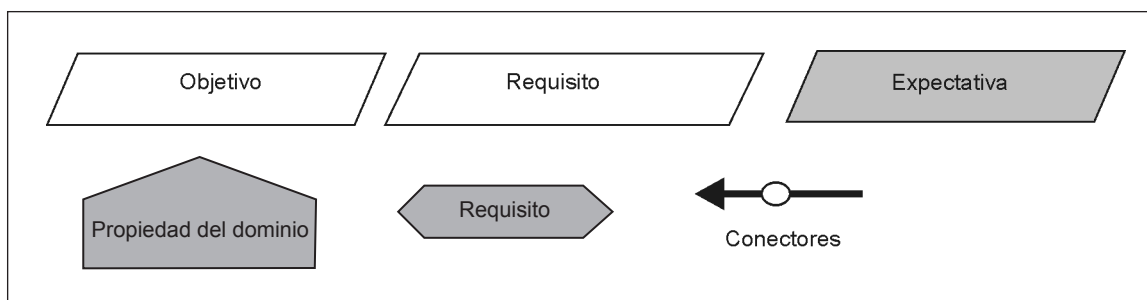


Figura 1. Símbolos del diagrama de objetivos de KAOS

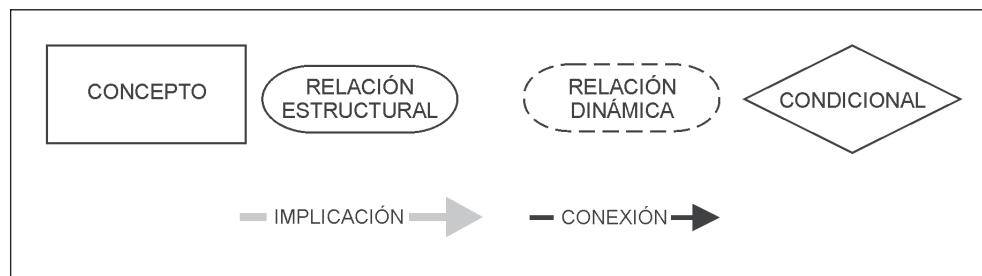


Figura 2. Símbolos del esquema preconceptual

diferentes esquemas conceptuales; es decir, los esquemas preconceptuales son representaciones del modelo verbal del problema que no presentan las ambigüedades propias del lenguaje natural y que se pueden obtener a partir del lenguaje controlado denominado UN-Lencep, cercano al lenguaje natural que emplea el interesado durante la fase de definición. En la figura 2 se pueden observar los elementos de este esquema, cuya descripción es la siguiente:

- Los conceptos. Son sustantivos o sintagmas nominales del discurso del interesado.
- Relación estructural: Es una relación permanente entre los conceptos. Está asociada con los verbos “es” y “tiene”.
- Relación dinámica. Está asociada con las operaciones definidas en Jaramillo *et al.* (2005). Por lo general, las operaciones son verbos que denotan actividades, como “registrar”, “pagar”, “elaborar”, etc.
- Implicación. Sirve para unir relaciones dinámicas o para unir condicionales con relaciones dinámicas, estableciendo entre ellas una relación causa-efecto.
- Condicional. Es una relación de causalidad que indica las restricciones o reglas del negocio que se deben cumplir.
- Conexión. Permite enlazar los conceptos con las relaciones y las relaciones con los conceptos.

El esquema preconceptual sólo prevé el uso de relaciones de tipo estructural y dinámico. Exis-

ten algunos verbos que identifican objetivos que no pertenecen a ninguna de las dos clasificaciones anteriores. Por ello, en Lezcano (2007) se propone un nuevo tipo de relación, denominada “relación de logro”, cuya representación gráfica puede observarse en la figura 3. Algunos de los verbos que se admiten en estas relaciones son los siguientes: avalar, garantizar, gestionar, incrementar, lograr, obtener, fomentar, promover, hacer.



Figura 3. Símbolo de relación de logro

La relación de logro puede tener una relación directa con las relaciones estructurales, las relaciones dinámicas y los conceptos. Para conectar estos elementos se utiliza una flecha punteada cuyo punto inicial es la relación de logro y cuyo punto final es la relación estructural, la relación dinámica o el concepto. En la figura 4 se puede observar este símbolo de conexión. Las relaciones de logro pueden ligarse entre sí mediante implicaciones, definiendo de esta manera el carácter jerárquico que se traduce luego al diagrama de objetivos de KAOS.

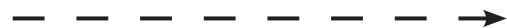


Figura 4. Símbolo de conexión de relaciones de logro



3. VALIDACIÓN DEL MÉTODO PARA LA OBTENCIÓN AUTOMÁTICA DEL DIAGRAMA DE OBJETIVOS

La validación se realiza mediante tres casos de estudio presentados por Letier (2001), Heaven y Finkelstein (2004) y Zapata *et al.* (2006d), respectivamente. Para cada uno de los casos, se traza el diagrama de objetivos de KAOS empleando el método descrito por Lezcano (2007) y después se compara el diagrama resultante con los diagramas respectivos que se proponen en los artículos. La comparación se realiza tomando en cuenta las siguientes características (Zowghi y Gervasi, 2002):

- **Compleitud.** Número de elementos de los diagramas. Cabe anotar que, por el nivel de abstracción en que se ubica el diagrama de objetivos, se debe tener naturaleza declarativa, ilustrando elementos que se centren en el “qué” del espacio del problema. La completitud de los diagramas no debe implicar naturaleza imperativa, involucrando elementos del “cómo” se piensa resolver. En este sentido, la completitud se evalúa tomando como base los enunciados de los casos y el número de elementos que se genera en los diagramas obtenidos.
- **Corrección.** Uso adecuado de la sintaxis de diagrama.
- **Consistencia.** Adecuación con la especificación.

Los dos primeros casos de estudio corresponden a los trabajos de Letier (2001) y de Heaven y Finkelstein (2004) y se relacionan con el sistema de servicio de ambulancias de Londres y el controlador avanzado y automático de trenes desarrollado por el San Francisco BART (Bay Area Rapid Transit). Zapata *et al.* (2006d) presentan el tercer caso de estudio, que corresponde al despacho y cobro de pedidos en una pizzería. Cabe anotar que los diagramas que presentan los autores de los casos de estudio se obtuvieron de forma manual, en tanto

que en la propuesta de Lezcano (2007) se aplicaron las reglas para generar el diagrama de forma automática, lo cual constituye la primera diferencia para analizar en relación con el estado del tema. En las secciones siguientes se detallan estos casos y se realiza la validación.

3.1 Caso de estudio: sistema de servicio de ambulancia de Londres

En este caso de estudio, propuesto por Letier (2001), se aplica un método manual para elaborar el diagrama de objetivos, mediante la identificación de los objetivos preliminares o de alto nivel, el primero de los cuales se llama “intervención de la ambulancia”. Luego para identificar los objetivos subrogados se responde a la pregunta: “¿Cómo se puede lograr el objetivo de alto nivel?”. En tercer lugar, se realiza el refinamiento del objetivo de alto nivel mediante las definiciones formales de los objetivos identificados previamente. Por último, se deriva el diagrama completo de objetivos. Heaven y Finkelstein (2004) retoman el trabajo realizado por Letier (2001) y proponen la representación del modelo de KAOS utilizando UML, representando los objetivos y los agentes por medio de los objetos del diagrama de colaboración de UML 1.4 (diagrama actual de comunicación de la versión 2.0), diferenciándolos con etiquetas. Cabe anotar que el proceso es completamente manual y que los diagramas obtenidos por los autores no guardan mucha relación con el enunciado presentado.

El enunciado de este caso de estudio propone que, para cada llamada urgente que reporte un accidente, deberá haber una ambulancia que arribe al lugar del incidente dentro de los 14 minutos siguientes. Cada vez que el despachador reciba el reporte de un accidente, debe enviar la ambulancia que esté más cerca del lugar del accidente; una vez llegue al lugar, el personal de ambulancia deberá intervenir y diligenciar un formulario de accidente. El despachador es el encargado de enviar la ambulancia, si

ésta se encuentra en la estación; de lo contrario, el despachador deberá entregar toda la información del accidente al radiooperador para que él localice y envíe la ambulancia (Letier, 2001).

3.1.1 *Diagrama de objetivos obtenido por Letier y por Heaven y Finkelstein*

Los diagramas de objetivos obtenidos por Letier (2001) y por Heaven y Finkelstein (2004) se pueden observar en las figuras 5 y 6 respectivamente.

3.1.2 *Diagrama de objetivos obtenido con el método de la Escuela de Sistemas*

El primer paso que se debe realizar para obtener el diagrama de objetivos consiste en la representación del modelo verbal en esquemas preconceptuales. Luego, al esquema preconceptual se aplican las reglas definidas en Lezcano (2007). Las figuras 7 y 8 muestran el esquema preconceptual y el diagrama de objetivos obtenidos con este método.

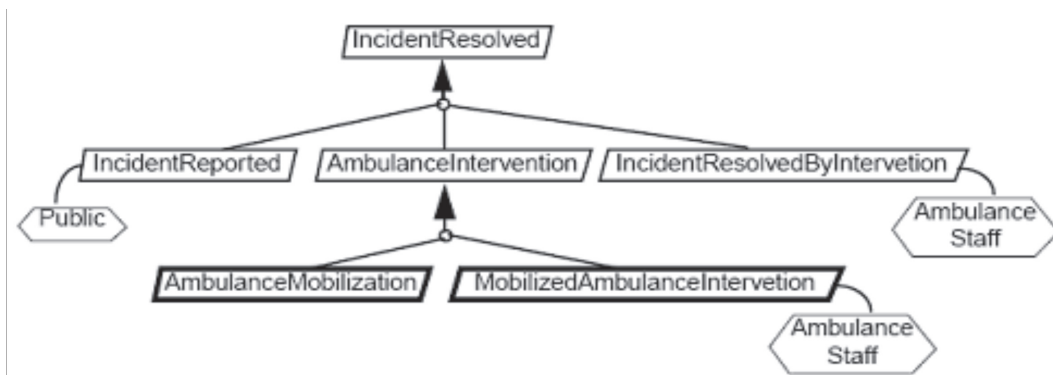


Figura 5. Diagrama de objetivos de Letier (2001) para el caso de la ambulancia

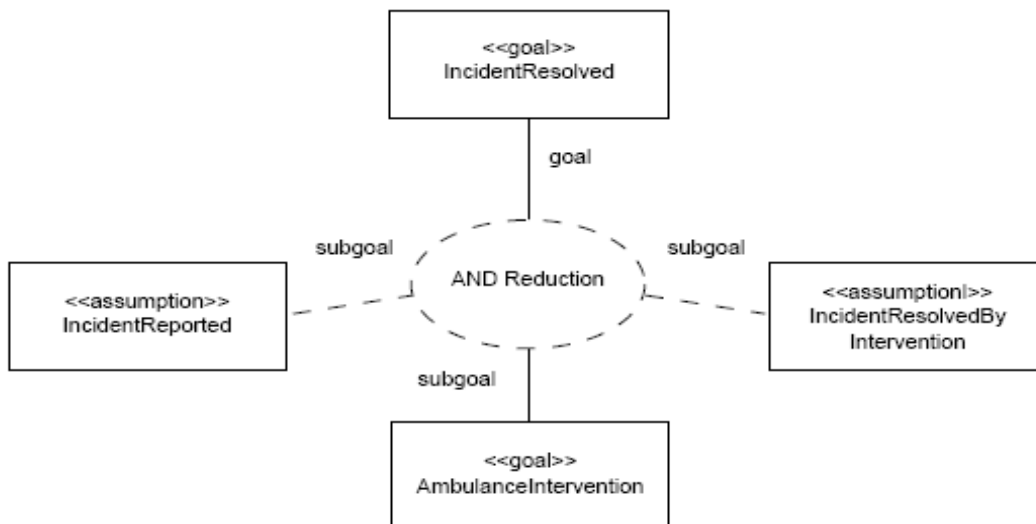


Figura 6. Diagrama de objetivos de Heaven y Finkelstein (2004) para el caso de la ambulancia

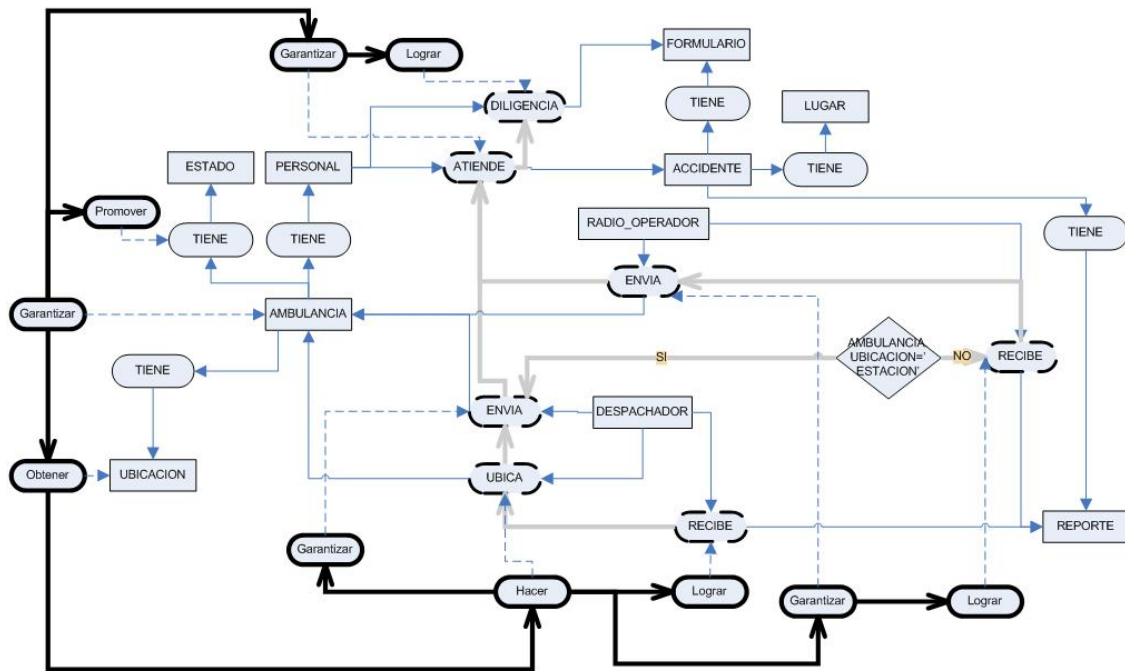


Figura 7. Esquema preconceptual del caso de la ambulancia

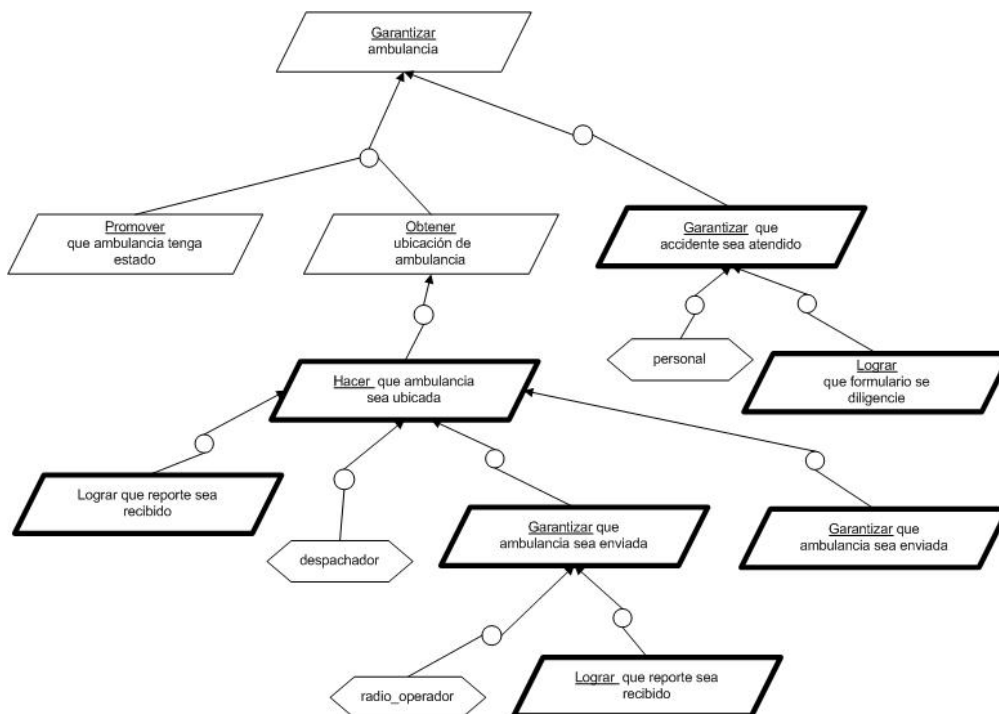


Figura 8. Diagrama de objetivos del caso de la ambulancia

3.1.3 Análisis de resultados

Al evaluar la completitud entre los diagramas obtenidos mediante el método de la Escuela de Sistemas y el obtenido por Letier (2001), se concluye que el primero es más completo debido a las razones siguientes:

- Los objetivos identificados en este artículo son: garantizar ambulancia, promover que ambulancia tenga estado y hacer que ambulancia tenga localización; mientras que los obtenidos por Letier (2001) son: incidente resuelto, incidente reportado, intervención de la ambulancia, incidente resuelto por intervención. Aunque en Letier (2001) se obtienen cuatro objetivos, existen dos objetivos similares que se pueden representar en uno solo.
- El diagrama de la Escuela de Sistemas obtiene los siguientes requisitos: hacer que ambulancia sea localizada, lograr que reporte sea recibido, garantizar que ambulancia sea enviada, garantizar que accidente sea atendido, lograr que formulario sea diligenciado; es decir, se identifican cinco requisitos; por otra parte, en Letier (2001) sólo se identifican dos requisitos: movilización de ambulancia e intervención de ambulancia movilizada.
- Los actores identificados por Letier (2001) son público y personal de ambulancia; mientras que en el diagrama de la Escuela de Sistemas se identifican: despachador, radiooperador y personal de ambulancia.

En conclusión, respecto a la completitud del diagrama de objetivos, con el método de la Escuela de Sistemas se identifica un mayor número de actores y requisitos que en Letier (2001) y Heaven y Finkelstein (2004) e igual número de objetivos.

Al evaluar el uso adecuado de la sintaxis, se tiene que Letier (2001) asocia los actores con objetivos y con requisitos; esto contradice la sintaxis utilizada en la metodología KAOS, que establece que los actores están asociados a los requisitos o expec-

tativas. Además, la jerarquización de los requisitos no es coherente con la metodología KAOS, al identificar requisitos subrogados en un mismo nivel, por ejemplo: movilización de ambulancia e intervención de la ambulancia movilizada; para llevar a cabo la intervención de la ambulancia movilizada primero se debe movilizar.

Los diagramas obtenidos en ambos trabajos presentan diferencias de consistencia con el modelo verbal presentado por el interesado. Sin embargo, los actores identificados por el diagrama de la Escuela de Sistemas corresponden al modelo verbal, mientras que algunos de los actores identificados en Letier (2001) no corresponden a dicho modelo, por ejemplo: público. Por otra parte, la representación del modelo verbal en los esquemas preconceptuales incluye la participación del analista en la determinación de los verbos correspondientes a las relaciones de logro. Por ello, esta intervención se refleja en el diagrama de objetivos, provocando que los objetivos y requisitos obtenidos se diferencien del modelo verbal únicamente en este tipo de verbos. Por ejemplo, en el objetivo “hacer que ambulancia tenga ubicación”; el verbo hacer es ingresado por el analista, pero la frase “ambulancia tiene ubicación” está incluida en el modelo verbal.

Otra de las fallas encontradas en Letier (2001) es que no se realiza una clara diferencia entre un objetivo, una operación y un estado, ya que los objetivos y requisitos que presenta tienen la forma de operaciones o de estados. Por ejemplo, “AmbulanceMobilization” tiene forma de operación, en tanto que “IncidentResolved” parece ser un estado.

Al realizar la comparación con el diagrama obtenido por Heaven y Finkelstein (2004) y el diagrama de la Escuela de Sistemas, se tiene que aquel presenta las siguientes deficiencias:

- En el diagrama no se identifican todos los elementos que hacen parte del Diagrama de Objetivos de la metodología KAOS, como son los actores y los requisitos. Además, sólo identifica cuatro



objetivos, de los cuales dos están etiquetados como objetivos y los otros dos como suposiciones. Es preciso anotar que la etiqueta “suposición” no tiene correspondencia con ningún elemento de la metodología KAOS. Por otra parte, los niveles jerárquicos identificados en ese trabajo son muy pocos (dos), en tanto que en el diagrama de la Escuela de Sistemas se tienen cinco niveles. Esto se presenta debido a que en este trabajo, además de identificar los objetivos, también se identifican requisitos y actores.

- Aunque Heaven y Finkelstein (2004) obtienen algunos elementos del diagrama de objetivos de KAOS, la sintaxis empleada es completamente diferente a la definida en la metodología KAOS.
- El trabajo de Heaven y Finkelstein (2004) presenta las mismas fallas del trabajo de Letier (2001) referentes a la consistencia con el modelo verbal y el uso inadecuado de objetivos y operaciones.

3.2 Caso de estudio: el controlador avanzado y automático de trenes

Letier (2001) realiza para este caso de estudio el mismo proceso utilizado para el de la ambulancia. En Heaven y Finkelstein (2004) ocurre un proceso similar al del caso de la ambulancia para el del controlador avanzado y automático de trenes.

En este caso de estudio, se toman en consideración los aspectos necesarios de un sistema para el control y aceleración de los trenes. El problema está dirigido al desarrollo de un sistema de control encargado de la aceleración y velocidad para obtener que los trenes corran rápida y suavemente, con las siguientes restricciones de seguridad:

- Un tren no deberá entrar en una puerta cerrada (en el sistema BART, una puerta cerrada no es un objeto físico, sino una señal recibida por el sistema de control de aceleración y velocidad, que establece si se permite o no el ingreso de un tren a un tramo de la vía).

- Un tren nunca debe ir cerca de otro, porque si el tren que está delante de repente se detiene (quizás debido a un descarrilamiento), el de atrás lo golpearía.

3.2.1 Diagrama de objetivos obtenido por Letier y por Heaven y Finkelstein

La metodología utilizada por Letier (2001) y por Heaven y Finkelstein (2004) para obtener el diagrama de objetivos del sistema controlador avanzado y automático de trenes desarrollado por el San Francisco BART es similar a la metodología utilizada para el servicio de ambulancia de Londres. En la figura 9 se puede ver el diagrama de objetivos obtenido por Letier (2001) y en la figura 10 puede observarse el diagrama obtenido por Heaven y Finkelstein (2004).

3.2.2 Diagrama de objetivos obtenido con el método de la Escuela de Sistemas

Para obtener el diagrama de objetivos con este método, se representa inicialmente el modelo verbal en el esquema preconceptual (figura 11). Posteriormente, se obtiene el diagrama de objetivos con la aplicación de las reglas presentadas en Lezcano (2007). Este diagrama se puede observar en la figura 12.

3.2.3 Análisis de resultados

Al realizar la comparación entre los diagramas de objetivos presentados por Letier (2001) y por Heaven y Finkelstein (2004) contra el diagrama elaborado con el método de la Escuela de Sistemas, se obtiene lo siguiente:

- En el diagrama obtenido por Letier (2001) se identifican cuatro objetivos y un requisito; además, no se identifican actores. Heaven y Finkelstein (2004) identifican cinco objetivos sin identificar requisitos y actores. En el diagrama de la Escuela de Sistemas, se identifican un objetivo, cinco requisitos y un actor. Por ello, puede concluirse que el diagrama de objetivos de la

Escuela de Sistemas identifica la mayor cantidad de elementos.

- Letier (2001) obtiene tres niveles de jerarquía; en el último nivel, presenta objetivos y requisitos, lo cual contradice el principio de completitud del diagrama de objetivos de KAOS que establece que en los nodos hoja (los de nivel más bajo en el diagrama) sólo pueden existir requisitos o expectativas. Por otra parte, la sintaxis empleada
- En Letier (2001) y Heaven y Finkelstein (2004) no se alcanza a observar la diferencia entre objetivos y operaciones, por ejemplo, en el objetivo “tren entrando en la puerta cerrada” se puede observar que se hace referencia a una operación o a un estado, debido a que no se conoce el momento en que concluye el desplazamiento.

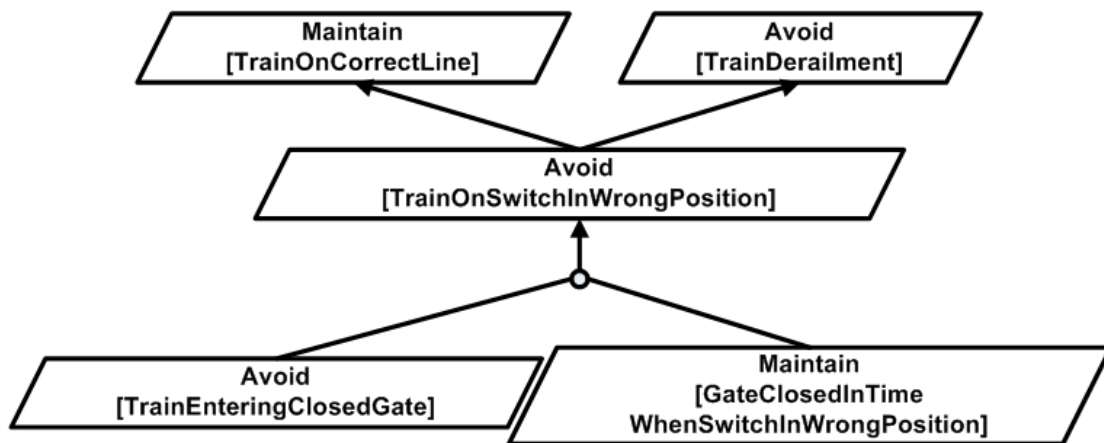


Figura 9. Diagrama de objetivos de Letier (2001) para el caso del tren

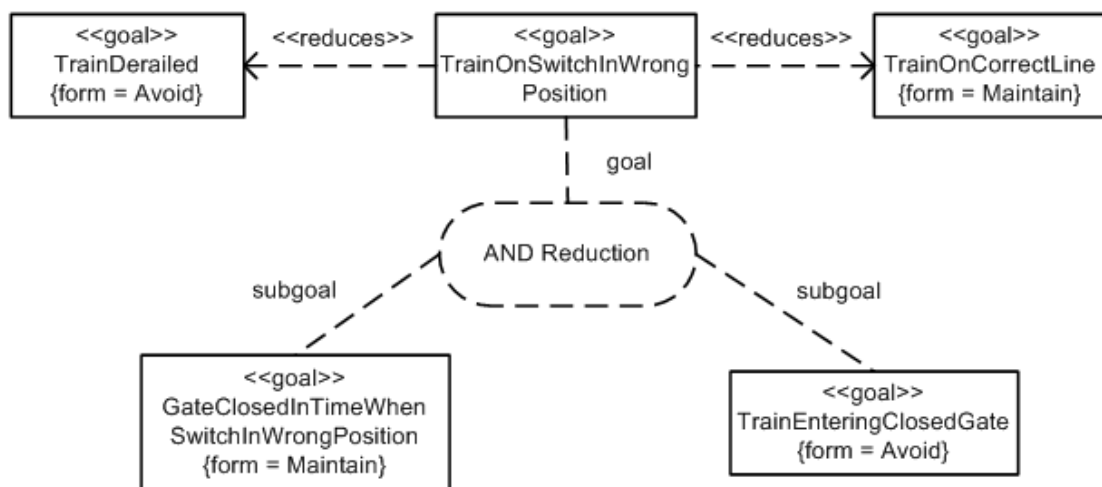


Figura 10. Diagrama de objetivos de Heaven y Finkelstein (2004) para el caso del tren

- En ninguno de los diagramas obtenidos por estos autores se presenta consistencia con el modelo verbal. Sin embargo, en el diagrama de objetivos de la Escuela de Sistemas, los actores identificados corresponden al modelo verbal, mientras que los objetivos y los requisitos identificados sólo se diferencian del modelo verbal en el verbo que se utiliza en la relación de logro.

3.3 Caso de estudio: Pizzería

Zapata *et al.* (2006d) emplean el diagrama de objetivos de KAOS como uno de los artefactos que se deben elaborar en el método de desarrollo de *software* denominado *UN-METODO*. Se establece en dicho método que el analista debe construir los diferentes artefactos a partir del discurso del dominio que logre capturar en las reuniones con los interesados.

En el caso de la pizzería, el discurso correspondiente es el siguiente: “La pizzería lleva ya cerca de un año de actividad y hasta ahora no se ha conseguido ganar tanto dinero como se había esperado. Esta

pizzería cuenta con un despachador, un cocinero y dos repartidores y ofrece a los clientes diversos tipos y tamaños de pizza, además de la posibilidad de ordenar suplementos. La pizzería tiene una zona de cobertura determinada, y, con el fin de competir frente a otras en la zona, se ha estado ofreciendo al cliente una promoción que consiste en entregarle el pedido gratis si tarda en ser entregado más de 30 minutos. Esta promoción ha atraído buena cantidad de clientes, pero también se ha convertido en una de las principales fuentes de pérdida de dinero, puesto que muy a menudo los repartidores no consiguen llevar todas las pizzas a tiempo. Con la inconformidad de los repartidores, se ha impuesto en el reglamento que “cada pedido entregado tarde deberá ser pagado por el repartidor responsable, con el fin de evitar tanta pérdida de dinero y conseguir que los repartidores se esfuercen más en su trabajo”.

3.3.1 Diagrama de objetivos obtenido por Zapata *et al.*

En la figura 13 se muestra el diagrama de objetivos obtenido por Zapata *et al.* (2006d) para el caso de estudio de la pizzería.

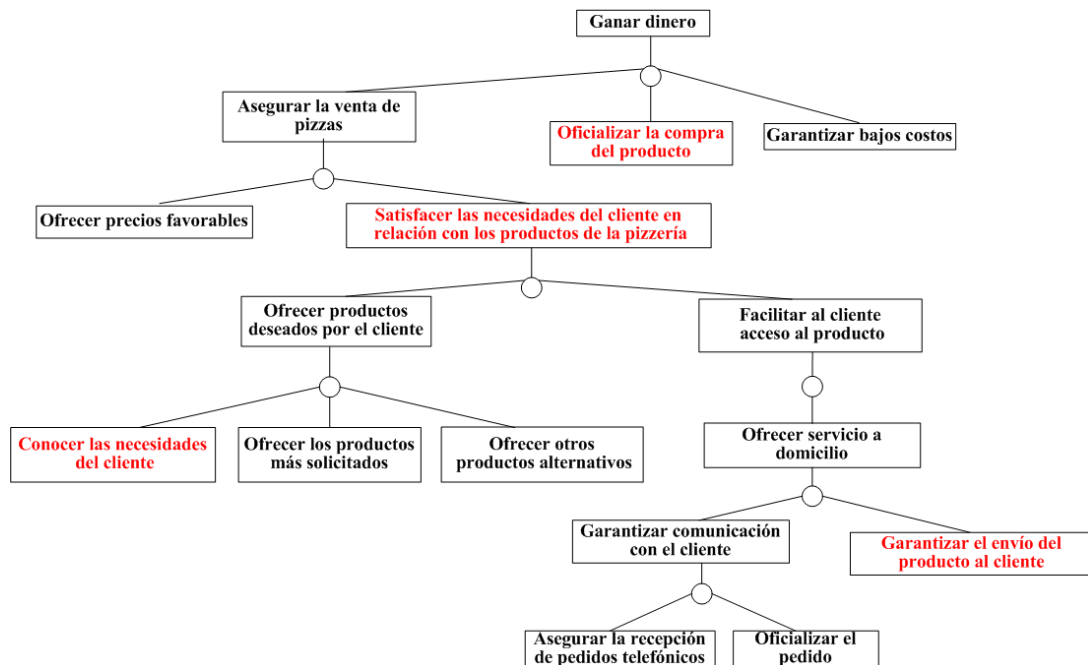


Figura 13. Diagrama de objetivos del caso de la pizzería



3.2.2 Diagrama de objetivos obtenido con el método de la Escuela de Sistemas

A partir del esquema preconceptual obtenido para el caso de la pizzería (figura 14) se obtuvo el diagrama de objetivos que se muestra en la figura 15.

3.3.3 Análisis de resultados

A continuación se describen las principales diferencias que se obtuvieron al realizar el proceso de validación mediante la comparación del diagrama obtenido por Zapata *et al.* (2006d) y el correspondiente a la Escuela de Sistemas.

- En el diagrama de Zapata *et al.* (2006d), se identifican 16 objetivos y no se logran identificar requisitos ni actores. Por otra parte, el diagrama de objetivos de la Escuela de Sistemas incluye dos objetivos, cinco requisitos y tres actores, lo que lleva a concluir que identifica mayor variedad de elementos.

- Los objetivos obtenidos por Zapata *et al.* (2006d) se identifican completamente mediante la experiencia del analista. Por ejemplo, el objetivo “ofrecer precios favorables” no se encuentra descrito en el modelo verbal.
- Los actores incluidos en el diagrama de la Escuela de Sistemas se extraen del modelo verbal; mientras que los objetivos y los requisitos presentan una leve diferencia con el modelo verbal consistente en la adición de las relaciones de logro que identifica el analista. Cabe anotar que ambos diagramas emplean verbos de logro para definir objetivos y requisitos, por lo cual no se presentan confusiones en relación con los objetivos y las operaciones y estados.
- Los dos diagramas de objetivos que se presentan para este caso de estudio utilizan de manera adecuada la sintaxis del diagrama de objetivos de KAOS.

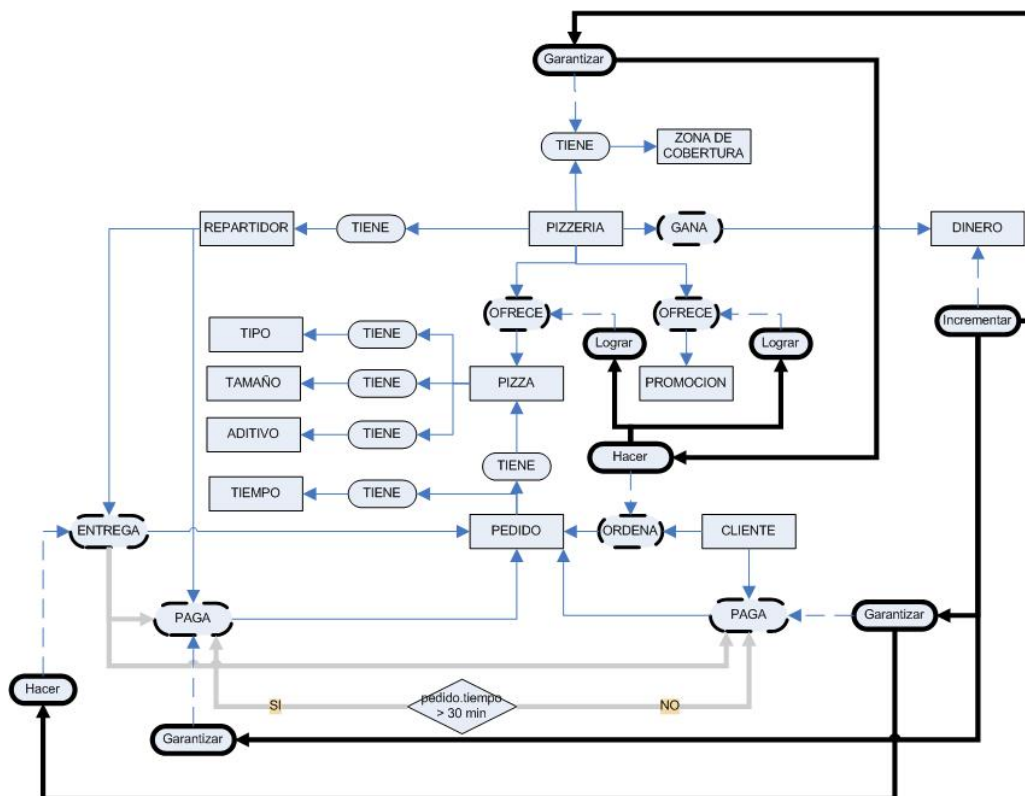


Figura 14. Esquema preconceptual del caso de la pizzería

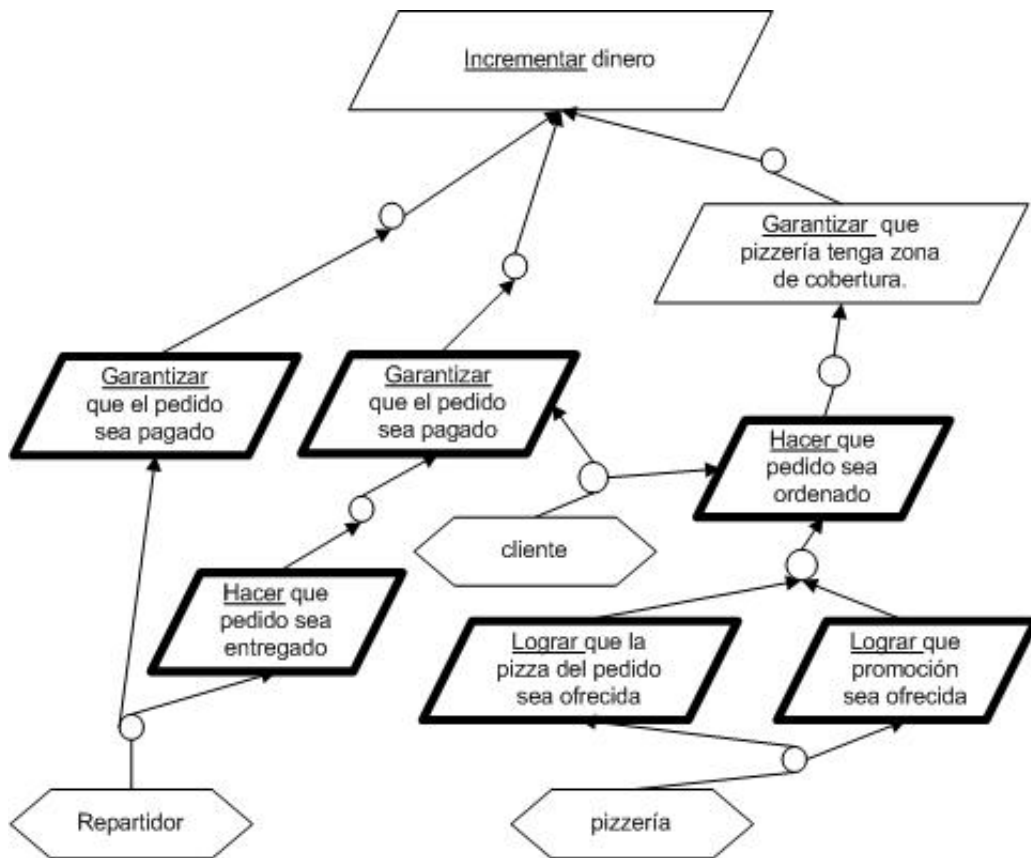


Figura 15. Diagrama de objetivos del caso de la pizzería

4. CONCLUSIONES

El diagrama de objetivos de KAOS es importante en el desarrollo de *software* porque permite enlazar los objetivos de alto nivel con los requisitos y expectativas de los interesados. Su construcción se realiza durante las fases iniciales del ciclo de vida del *software* y, por lo general, se hace de forma manual, lo cual genera problemas de uso de su sintaxis, falta de adecuación del diagrama con el discurso del interesado y dificultades inherentes a la alta subjetividad del analista en su construcción. Estos problemas motivaron el método para la generación automática del diagrama de objetivos a partir de los esquemas preconceptuales, que se presenta en Lezcano (2007).

Con la validación de dicho método, que se presenta en este artículo, se demostró que, a partir de los esquemas preconceptuales, logra obtenerse el diagrama de objetivos de KAOS de acuerdo con el modelo verbal del interesado, minimizando la participación del analista en la identificación de los elementos del diagrama y su construcción. En los diferentes casos de estudio, las características de completitud, corrección y consistencia fueron mejores que las obtenidas para el diagrama de objetivos elaborado en otros trabajos.

5. TRABAJOS FUTUROS

A partir de este trabajo, se generan nuevos asuntos que pueden suministrar continuidad al



realizado para la elaboración del diagrama de objetivo, como los que se enuncian enseguida.

- Identificar los demás elementos del diagrama de objetivos: expectativas y propiedades del dominio.
- Incluir los lineamientos teóricos de este artículo en una herramienta CASE que permita la construcción automática o semiautomática del diagrama de objetivos. En este caso, la interoperabilidad que se logra con los lenguajes tipo XMI y la cercanía de este lenguaje con una descripción de tipo ontológico en lenguajes como OWL generan un interesante campo de acción, que podría combinar las ontologías y los esquemas preconceptuales para generar un estándar de comunicación entre herramientas CASE para la generación automática de los diferentes diagramas necesarios en el ciclo de vida del *software*.
- Definir reglas adicionales que puedan generar variaciones sobre los elementos identificados en el diagrama de objetivos.
- Definir una nueva opción para la identificación de objetivos, teniendo en cuenta el participio pasado de los verbos, utilizado por algunos autores como Letier (2001) y Heaven y Finkelstein (2004).

BIBLIOGRAFÍA

1. HEAVEN, W. and FINKELSTEIN, A. (2004). UML profile to support requirements engineering with KAOS. *IEEE Proceedings Software*. Vol. 151; Part 1. p. 10-27.
2. JARAMILLO, A.; ZAPATA, C. y ARANGO, F. (2005) Una propuesta para el reconocimiento semiautomático de operaciones utilizando un enfoque lingüístico. En: *Revista Facultad de Ingeniería Universidad de Antioquia*. No 34. p. 42-51.
3. LAMSWEERDE, A.; DARDENNE, A. and FICHAS S. (1993). Goal-directed requirements acquisition. En: *Science of Computer Programming*, Vol. 20. p. 3-50.
4. LETIER, E. (2001). Reasoning about agents in goal-oriented. PhD. Thesis. Louvain: Département d'Ingénierie Informatique. Université Catholique de Louvain.
5. LEZCANO, L. (2007). Elaboración semiautomática del diagrama de objetivos a partir de lenguaje natural restringido. M.Sc. Tesis. Medellín: Universidad Nacional de Colombia.
6. ORACLE Corporation. (2000). Oracle method CDM quick tour. Oracle Corporation, Redwood City.
7. ZAPATA, C.; GELBUKH, A. and ARANGO, F. (2006a). Pre-conceptual schema: a UML isomorphism for automatically obtaining UML conceptual schemas. *Research in Computing Science: Advances in Computer Science and Engineering*, Vol. 19. p. 3-13.
8. ZAPATA, C.; GELBUKH, A. y ARANGO, F. (2006b). UN-Lencep: Obtención automática de diagramas UML a partir de un lenguaje controlado. 3er Taller de Tecnologías del Lenguaje Humano. Encuentro Nacional de Ciencias de la Computación, San Luis Potosí, Septiembre 2006.
9. ZAPATA, C.; GELBUKH, A. and ARANGO, F. (2006c). Pre-conceptual schema: A conceptual-graph-like knowledge representation for requirements elicitation. *Lecture Notes in Computer Science*, Vol. 4293. p. 17-27.
10. ZAPATA, C. M.; VILLEGAS, S. y ARANGO F. (2006d). Reglas de consistencia entre modelos de requisitos de UN-Metodo. En: *Revista Universidad Eafit*. Vol 42. No 141. p. 40-59.
11. ZOWGHI, D. and GERVASI, V. (2002). The three Cs of requirements: consistency, completeness and correctness. En: *Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ'02)*, Essen, p. 155-164.