

DEFINICIÓN DE UN MÉTODO PARA REALIZAR LA EDUCCIÓN DE REQUISITOS DE SOFTWARE A PARTIR DE LA ARTICULACIÓN DEL DIAGRAMA CAUSA EFECTO CON EL DIAGRAMA DE OBJETIVOS DE KAOS

Luis Alfonso Lezcano Rodríguez¹, Leidy Biviana Herrera Méndez², Ana Milena Londoño Agudelo³

¹Ingeniero de Sistemas, Magíster en Ingeniería de Sistemas y (c) PhD en Ingeniería con énfasis en Sistemas. Profesor de cátedra, Departamento de Sistemas, Facultad de Ingeniería, Instituto Tecnológico Metropolitano ITM, Carrera 31 N° 54-10- Medellín, luislezcano5161@correo.itm.edu.co.

²Estudiante de ingeniería de Sistemas. Instituto Tecnológico Metropolitano ITM, Carrera 31 N° 54-10- Medellín, leidyherrera55136@correo.itm.edu.co.

³Estudiante de ingeniería de Sistemas. Instituto Tecnológico Metropolitano ITM, Carrera 31 N° 54-10- Medellín, analondono56175@correo.itm.edu.co

RESUMEN

El diagrama causa efecto y el diagrama de objetivos de KAOS son utilizados en la fase de definición y análisis de requisitos de software para reducir la brecha de comunicación entre analista y usuario. En la actualidad persisten algunos problemas durante el desarrollo del software como son: (i) la comunicación limitada, y (ii) la falta de validación de requisitos. Además, no existe una caracterización desde el lenguaje natural de los elementos básicos que componen el diagrama causa-efecto y el diagrama de objetivos de KAOS. Estos problemas no permiten una correcta identificación y validación de los requisitos para el desarrollo del software haciendo que se presenten correcciones en las etapas de implementación y producción del software. Este artículo propone (i) caracterizar desde el lenguaje natural los elementos básicos del diagrama causa efecto; (ii) los elementos básicos del diagrama de objetivos de KAOS; y (iii) articular el diagrama de objetivos de KAOS con el diagrama causa efecto.

Palabras clave: Diagrama de Objetivos de KAOS, diagrama causa efecto, Ingeniería del software, educación de requisitos.

Recibido: 4 de diciembre de 2013.
Received: December 4th, 2013.

Aceptado: 12 de Junio de 2013.
Accepted: June 12th, 2013.

DEFINITION OF A METHOD TO PERFORM SOFTWARE REQUIREMENTS ELICITATION FROM THE JOINT OF CAUSE EFFECT DIAGRAM WITH KAOS GOAL DIAGRAM

ABSTRACT

The KAOS goal diagram and cause-effect diagram are used in the definition phase and analysis of software requirements to reduce the communication gap between analyst and stakeholder. At present, there are still some problems during software development, such as: (i) limited communication between stakeholder and the analyst, and (ii) the lack of validation of requirements between the analyst and the stakeholder. There is no natural language characterization of the basic elements making up the KAOS goal diagram and cause-effect diagram. These issues avoid the proper identification and validation of requirements for the software development, producing multiple corrections in the software implementation and production stages. This paper proposes (i) to characterize natural language from the basics of KAOS goal diagram; (ii) the basic elements of the cause and effect diagram; and (iii) articulate the cause and effect diagram with the KAOS goal diagram.

Keywords: KAOS Goal Diagram, Cause and Effect Diagram, Software Engineering, Requirements Elicitation.

1. INTRODUCCIÓN

La Ingeniería de Requisitos tiene como propósito hacer que los requisitos de una pieza de software logren alcanzar un grado óptimo antes de comenzar con la fase de diseño del proyecto. Éstos deben ser medibles, comprobables y sin ambigüedades. La educación de requisitos es la etapa más importante dentro de la fase de desarrollo de un software. Uno de los desafíos más importantes en esta etapa es asegurar que los requisitos sean consistentes con las expectativas y necesidades del cliente. Es por esto que se hace necesario conocer sobre la organización a través de entrevistas con los clientes y usuarios para que planteen los problemas a resolver y con esta información el analista pueda hacer uso de algunas herramientas como lo es el diagrama de objetivos de KAOS y el diagrama causa-efecto.

El diagrama de objetivos tiene varios usos: (i) liga los requisitos y expectativas del interesado con los objetivos de la organización, (ii) establece las responsabilidades de los actores; la espina de pescado también conocido como el diagrama causa-efecto, es una herramienta que permite identificar las posibles causas que generan un problema. Sin embargo, actualmente este diagrama (causa-efecto) no se encuentra articulado con el diagrama de objetivos de KAOS. En los diferentes trabajos en que se especifican o se usa el diagrama de objetivos de KAOS y el diagrama causa-efecto, subsisten algunos problemas tales como: el analista es, por lo general, quien elabora los diagramas de forma manual y subjetiva, ello implica que se presenten confusiones terminológicas y no se logren caracterizar desde el lenguaje natural los elementos que componen dichos diagramas.

Entre las metodologías de desarrollo que la Ingeniería de Software utiliza, se encuentra el RUP (Rational Unified Process), que emplea un conjunto de actividades para transformar las especificaciones de los requisitos de un interesado en un sistema. En este proceso se utiliza el Lenguaje Unificado de Modelamiento (UML) para elaborar los modelos conceptuales del sistema informático. El RUP tiene un ciclo de vida compuesto por las siguientes fases: Definición, Análisis, Diseño, Construcción, Transición, Producción. La fase de definición es aquella donde se detallan los objetivos a alcanzar o las metas a cumplir en el desarrollo del producto final. En la

fase de análisis se debe tener claro cuál es la problemática a resolver, para esto es esencial que el diagrama de objetivos de KAOS y el diagrama causa-efecto logren plasmar las necesidades y/o requisitos del interesado. Este trabajo se centrará en las dos primeras fases del ciclo de vida del software. El artículo tiene la siguiente estructura: 2 presenta el marco teórico y lo referente a Ingeniería del Software, Ingeniería de Requisitos y educación de requisitos, en la 3 están los antecedentes, en la 4 está el método propuesto en donde se definen algunos conceptos básicos y las reglas gramaticales para la construcción del diagrama causa-efecto, y de los elementos básicos del diagrama de objetivos de KAOS, en la 5 los resultados y, finalmente, el trabajo futuro y las conclusiones.

2. MARCO TEORICO

2.1 Ingeniería de Software

En [1], indican que la Ingeniería de Software se basa en las ciencias de la computación y la matemática, pero las ingenierías se amplían más allá de esta técnica para inspirarse en un rango de disciplinas. La Ingeniería de Software es diferente a las demás ingenierías, ya que no sólo se trata de lo intangible del software sino también de la naturaleza discreta del mismo. Las diferencias entre la Ingeniería del Software y las demás ingenierías son las siguientes: (i) las ciencias empleadas son de computación y no ciencias naturales; (ii) se trabaja en matemática discreta y no matemática continua; (iii) las entidades son abstractas o lógicas y no en artefactos físicos; (iv) en el sentido tradicional no existe una etapa de fabricación; y (v) el mantenimiento del software es un desarrollo continuado o de evolución.

En [2] se define la Ingeniería de Software como una asignatura de la ingeniería que abarca los aspectos de creación de software desde la etapa inicial que es la definición del sistema hasta su mantenimiento, una vez entra en producción. Existen dos frases claves en esta definición: (i) disciplina de la ingeniería: aquí los ingenieros a través de métodos y teorías usándolas de manera selectiva buscan soluciones a los problemas; así mismo deben trabajar con limitaciones financieras y organizacionales; (ii) todos los aspectos de producción de software: esta etapa abarca los procesos técnicos del desarrollo de software, además incluyen trabajos vinculados a la gestión

de proyectos de software, el desarrollo de herramientas, métodos y teorías de apoyo de producción de software.

2.2 Ingeniería de Requisitos

La Ingeniería de Requisitos (IR) incluye la primera etapa del ciclo de vida del software, ésta ha tenido gran importancia a partir de los años 70, se considera como una de las principales fases durante el desarrollo del software.

En [3], definen la Ingeniería de Requisitos como la aplicación de principios, métodos y técnicas para el descubrimiento de los requisitos de un producto de software, también permite el análisis, la documentación de los objetivos funciones y restricciones de dicho sistema. Indican como falencia la falta de acuerdo sobre los lenguajes, métodos y herramientas para su ejecución. Asimismo, hacen referencia a algunas herramientas CASE (Ingeniería del software asistida por computador), las cuales facilitan el diseño y la trazabilidad de un producto de software en los primeros pasos del ciclo de vida del software brindando soporte para el buen desarrollo del producto.

En [4], indican que la Ingeniería de Requisitos se deriva de la Ingeniería del Software que ayuda al analista de sistemas a traducir los objetivos en las funciones y restricciones de un sistema. También describen cada una de las etapas de la Ingeniería de Requisitos donde la primera se conoce como fase de definición, en ésta se identifica la problemática que se va afrontar, las causas que producen el problema y se afronta la posible solución. En la segunda fase, se plantean los requisitos, en la tercera se realiza la construcción de un documento en el cual se plantean las funcionalidades y restricciones del sistema a desarrollar, en la última fase, se realiza el modelo conceptual que contendrá la solución de acuerdo con los requisitos y las restricciones. En la figura 1 se pueden observar las fases que utiliza la Ingeniería de Requisitos.

En [5], plantean que la Ingeniería de Requisitos está comprendida por las etapas fundamentales del desarrollo en la Ingeniería del Software, la cual se enfoca en investigar, analizar, redactar y verificar los servicios y restricciones del sistema. Su mayor trascendencia es la de definición de los requisitos ya que de éstos depende el resultado de las etapas posteriores al desarrollo.



Figura 1. Fases de la Ingeniería de Requisitos. Fuente: González et al. (2008).

Si los requisitos son hallados o modificados en alguna de las etapas avanzadas del desarrollo esto provocará atrasos en el cronograma, aumento en el presupuesto y el riesgo de que el producto no satisfaga las necesidades del cliente.

2.3 Educción de Requisitos

En [6], se indica que la educción de requisitos es la etapa en la cual se recolecta, se procesa y se especifica la información que sirve para cumplir con los requisitos de los interesados. Para la especificación de los requisitos se usan varios diagramas, entre los cuales se encuentra el diagrama causa-efecto, éste ayuda a encontrar información importante para cumplir con las necesidades y requisitos de los interesados.

El diagrama causa efecto ayuda al analista y al interesado a comprender las causas que generan un problema, asimismo, sirve como apoyo en los procesos de educción y validación de requisitos de software. Además, permite: (i) identificar posibles soluciones; (ii) definir la toma de decisiones; y (iii) organizar planes de acción [7].

En [8] se indica que el diagrama causa-efecto fue realizado por Kaoru Ishikawa, el cual es un experto en dirección de empresas e interesado en mejorar el control de la calidad. Dicho diagrama se compone de: (i) un recuadro, (ii) una línea principal; y (iii) cuatro o más líneas que apuntan a la línea principal. En la figura 2 se puede observar la estructura del diagrama.

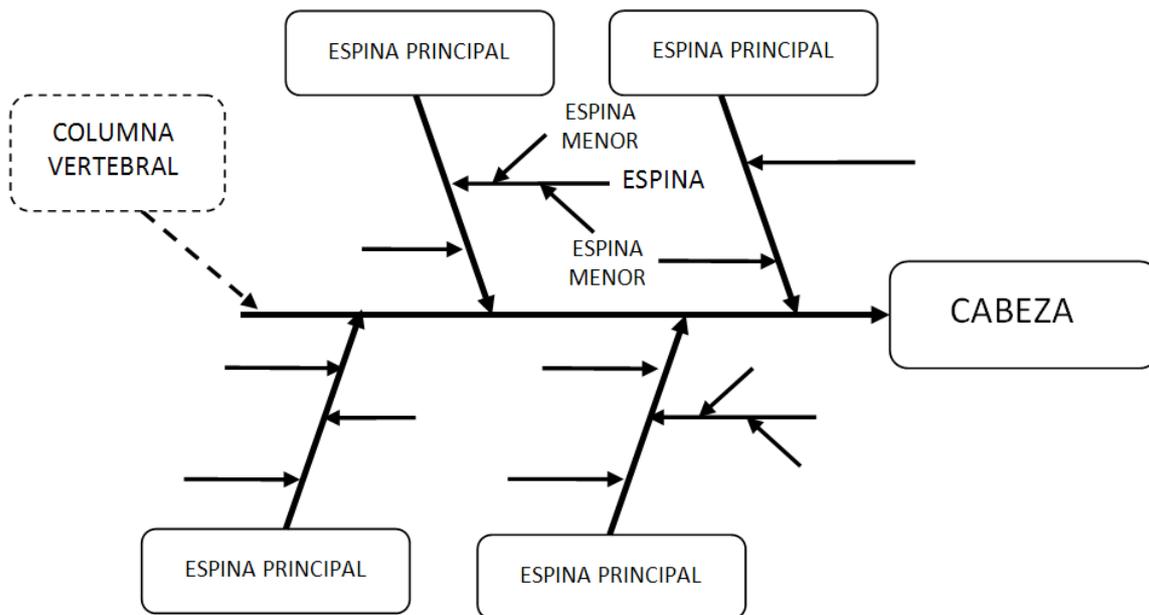


Figura 2. Estructura del diagrama de causa efecto. Fuente: Vargas 2010

Otro de los diagramas utilizados en la educación de requisitos es el diagrama de objetivos de KAOS, el cual se utiliza en la Ingeniería de Requisitos para la construcción de modelos e identificación de software.

En [9] se define el diagrama de objetivos de KAOS como una metodología en la cual, si se cumplen todos los sub objetivos, se logra cumplir con la condición para lograr el objetivo principal. Además, determina la representación de objetivos, agentes y objetos. Asimismo, presenta los siguientes pasos para la construcción del diagrama de objetivos: (i) Elaborar la estructura de objetivos, ésta se logra identificar mediante el propósito, las palabras claves, la intención, entre otras. Esta actividad se realiza a través de lectura inicial de la información suministrada por el interesado; (ii) Identificar los objetivos involucrados: cuando los objetivos están totalmente identificados se puede establecer sobre cual objeto del problema inciden; (iii) Identificar agentes y operaciones: los objetivos se refieren a transiciones de estados específicos, y por cada una de éstas se identifica la operación que la causa. Los agentes son identificados como los potenciales responsables de la operación; (iv) Ligar los objetivos para luego volverlos a requisitos del software aquí se consolida las condiciones del dominio y así se garantiza el cumplimiento de los objetivos asociados a cada operación.

En [10], se utilizan y se presentan algunos de los elementos que conforman el diagrama de objetivo de KAOS, los cuales se pueden observar en la figura 3. La definición de cada uno de estos elementos se presenta a continuación: (i) objetivo: solución a encontrar en un proceso o en la misión de una institución; (ii) requisito: meta que no es negociable dentro del proceso de desarrollo; (iii) expectativa: son objetivos que no se esperar agregar pero pueden ser incluidos si las condiciones del proyecto lo ameritan; (iv) actor: es la persona encargada de un requisito o una expectativa; (v) propiedad del dominio: son elementos requeridos para el cumplimiento de los objetivos; (vi) conectores: son flechas que enlazan objetivos, expectativas, requisitos, propiedades del dominio y actores. Pueden ser relaciones tipo "AND" cuando se solicita el cumplimiento simultáneo de varios objetivos que están unidos con conectores para que el sub objetivo se cumpla, o relaciones de tipo "OR" cuando se cumple uno de los objetivos unidos por el conector.

En [11], presentan la importancia de los objetivos y la manera como se definen y usan en las diferentes áreas del conocimiento, especialmente en el área de la Ingeniería de Software, además, caracterizarán las falencias actuales.

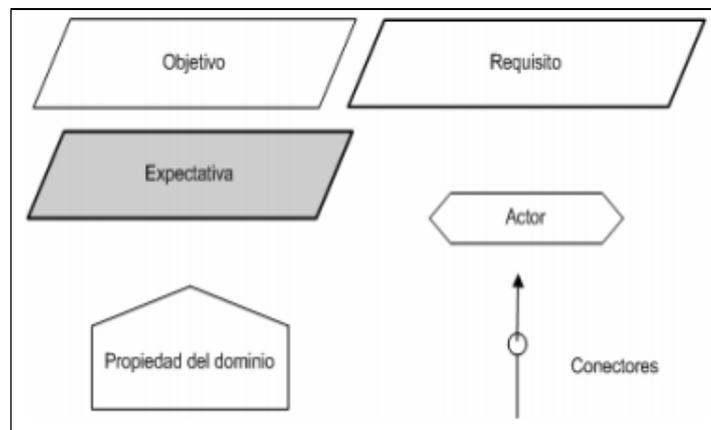


Figura 3. Elementos del diagrama de objetivos de KAOS. Fuente: Lezcano (2007).

2.4 Procesamiento del Lenguaje Natural.

En [12], definen el procesamiento de lenguaje natural como un campo de estudio y del desarrollo de la lingüística y de la Inteligencia Artificial, que se encarga de formular e investigar mecanismos que permitan procesar la comunicación entre humanos o entre humanos y máquinas a partir del lenguaje natural. El procesamiento de lenguaje natural cubre campos como el reconocimiento del habla, la traducción automática o la generación de lenguajes naturales que comprenden el análisis y comprensión del lenguaje y la extracción de información.

En [13] se define el procesamiento del lenguaje natural (PLN), el cual encierra un modelo del lenguaje natural en algoritmos adecuados y eficaces. Señala que las técnicas utilizadas para el modelado son relacionadas con eventos en diferentes campos donde se incluye la ciencia de la computación en la cual se suministra métodos para realizar algoritmos para las herramientas de software, la lingüística ayuda con procesos y modelos lingüísticos, modelos formales y métodos de la neurociencia en la cual investiga mecanismos mentales y otro tipo de actividades físicas. El procesamiento del lenguaje natural utiliza las computadoras para entender los diferentes lenguajes humanos. No se pretende decir que el computador es un ser humano que tiene pensamientos, conocimientos y sentimientos, sino que pueda reconocer y usar información expresada en el lenguaje humano.

Asimismo, el PLN está basado en modelos y métodos que se pueden clasificar en simbólicos, estadísticos, conexionistas y los enfoques híbridos. Los dos primeros son modelos matemáticos del

lenguaje natural. El enfoque simbólico se basa en el saber y emplean reglas y algoritmos que manejan estructuras de datos simbólicos que representan el conocimiento del lenguaje natural. El estadístico recopila muestras del lenguaje (corpus), éstas son marcadas para crear modelos estadísticos para el PLN. La técnica conexionista utiliza redes neuronales para describir el conocimiento lingüístico, y por último, las híbridas es la combinación de uno o más de los modelos anteriores con el propósito de integrar las ventajas de cada uno y resolver problemas de dominios y aplicaciones específicas.

En [14] se define el procesamiento del lenguaje natural como la habilidad de la máquina para procesar información comunicada, no sólo las letras y los sonidos del lenguaje. La ciencia que estudia PLN es llamada Lingüística Computacional.

Es importante resaltar que, la educación de requisitos de software requiere como insumo de entrada el universo del discurso expresado en lenguaje natural.

3. ANTECEDENTES

3.1 Diagrama Causa-Efecto

El diagrama causa-efecto está siendo aplicado en diferentes áreas del conocimiento entre éstas se encuentra: (i) la medicina, (ii) la producción, (iii) los sistemas; y (iv) el área de calidad.

En la medicina el diagrama causa efecto es utilizado para el análisis de casos, ya que permite evidenciar con claridad la relación que hay entre los problemas y las causas que lo están generando, también es utilizado para verificación de hechos

puntuales lo que ayuda a un mejor análisis, a buscar soluciones y a modificar procedimientos o métodos inadecuados. Para la construcción del diagrama causa-efecto realizan un listado de los factores potenciales que están causando la dificultad al usuario, luego, los factores son clasificados por categorías, consecutivamente se jerarquizan y finalmente se ubican en el diagrama [15].

En [16], utilizan el diagrama causa efecto en el campo de la medicina para aplicar un control de calidad en el tratamiento del dolor postoperatorio, partiendo de una información extraídas de tablas hechas en hojas de cálculo, y aunque lo hacen en lenguaje natural, no definen reglas gramaticales que puedan ayudar a la construcción de dicho diagrama desde un discurso presentado en lenguaje natural.

[17] aplica en su tesis doctoral el diagrama causa-efecto como una técnica para la construcción de indicadores en proyectos de ingeniería civil, partiendo de once objetivos propuestos en el proyecto de investigación LENSE (Proyecto Europeo del VI programa marco), los cuales son fundamentales en la construcción sostenible en Europa. La aplicación de este diagrama ha permitido un mejor análisis de adelante hacia atrás de los impactos potenciales para lograr un ambiente construido de acuerdo con el desarrollo sostenible. Aunque el diagrama causa-efecto es construido a partir del universo del discurso dado en lenguaje natural éste no tiene unos parámetros o reglas definidas que les permita identificar de manera consistente (entre el lenguaje natural y el diagrama obtenido) el problema y sus causas.

En [18], proponen el uso del diagrama causa-efecto en Ingeniería de Requisitos, indican que este diagrama debe tener en cuenta las metas de la organización, para argumentar la solución informática y las especificaciones para dicha solución. La construcción del diagrama causa-efecto consiste en asignarle de manera subjetiva porcentajes a las causas de tal manera que las sub causas que acompañan la principal no supere el 100%, de igual manera, la causa que posea el mayor porcentaje será considerada el problema principal a resolver conforme a las metas de la organización. Aunque definen algunas fórmulas para la asignación de dichos pesos, no definen reglas gramaticales que permitan establecer desde el lenguaje natural el problema principal y las

causas que lo generan. Además, el trabajo presentado utiliza el lenguaje controlado UN Lencep, el cual minimiza la expresividad en el universo del discurso.

3.2 Diagrama de objetivos de KAOS

En [9], se propone la búsqueda de palabras clave en los documentos de información disponibles para la identificación de objetivos. Así mismo, determina la representación de agentes, objetos, metas y la operacionalización de Metas/Objetivos en requisitos sobre objetos. Asimismo, presenta una propuesta para la representación del diagrama de objetivos, que se basa en la metodología KAOS (Knowledge Acquisition in Automated Specification). En este diagrama, se inicia con el objetivo de más alto nivel de la organización, el cual se subroga en otros objetivos hasta lograr los requisitos que debe cumplir una aplicación de software para satisfacer los objetivos organizacionales. El proceso para la realización del diagrama de objetivos de KAOS, necesita que se definan los objetivos secundarios que derivan los objetivos generales, para luego mostrar los objetivos más elementales que los derivan, y así sucesivamente hasta llegar a objetivos que no valen la pena descomponer o hasta que se alcancen expectativas, requisitos o propiedades del dominio. El diagrama de objetivos de KAOS se compone de los siguientes elementos: (i) objetivo; (ii) relación AND y OR; (iii) conflicto, (iv) expectativa, (v) requisito, (vi) propiedad del dominio; (vii) agente, (viii) relación de responsabilidad; y (ix) relación de asignación. En el diagrama de objetivos de KAOS aún persisten las siguientes falencias: (i) no existe una caracterización de los elementos básicos a partir del lenguaje natural; y (ii) no se articula con el diagrama causa-efecto.

En [19], [20] y [21], se plantea un método basado en heurísticas para identificar, clasificar, refinar y elaborar: objetivos, interesados, agentes y restricciones. Busca atacar los problemas originados durante el proceso de educación y especificación de requisitos (imprecisiones, ambigüedad, desacuerdos entre los interesados – usuarios). Para resolver estos problemas propone la utilización de un procedimiento que permita identificar, clasificar y refinar los objetivos que la aplicación de software debe alcanzar, con la subsiguiente transformación de esos objetivos en requisitos operacionales. Las actividades que debe realizar el analista al emplear esta propuesta, utilizando el método GBRAM (Goal-Based

Requirements Analysis Method), las cuales se integran en un modelo de: Entrada, Proceso y Salida, donde el Proceso se divide en Análisis y Refinamiento. Las principales falencias encontradas en este trabajo son: (i) no se utiliza el diagrama de objetivos; (ii) no caracteriza los elementos básicos del diagrama de objetivos de KAOS a partir del lenguaje natural; y (iii) no se articula con el diagrama causa efecto.

En [22], proponen un conjunto de reglas para la aplicación de varios métodos indicando el orden y la secuencia que se debe tener en cuenta para evitar ambigüedades o redundancias en la información, permitiendo un mejor manejo de los artefactos para el desarrollo del software, para esto crean la metodología “Un-Método”, que toma elementos de otros como “RUP” y “CDM”, e incorporan otros nuevos. Pese a incluir diagramas para el análisis de problemas y objetivos, el analista debe construirlos de manera subjetiva, lo cual genera incompatibilidades entre el universo del discurso presentado en lenguaje natural y los diagramas obtenidos.

En [23], proponen obtener el diagrama de objetivos a través de los esquemas preconceptuales teniendo en cuenta las especificaciones de consistencia, corrección y completitud, que permite minimizar los errores y disminuye la intervención del analista en la identificación y construcción del diagrama. Las falencias más representativas son: (i) no incluyen algunas palabras para calificar objetivos, tales como los adjetivos y los adverbios; (ii) no existe una estructura gramatical definida, (iii) no se plantean métodos automáticos que permitan relacionar objetivos y problemas.

En [8] se plantea un método y un prototipo que se basa en estructuras gramaticales que sirven para representar y enunciar los objetivos y los problemas en los diagramas causa-efecto y de objetivos de KAOS; además, propone unas reglas de consistencia que permiten establecer al momento de la construcción del diagrama de objetivos de KAOS y causa-efecto, la relación entre los objetivos y los problemas de estos diagramas. Aunque es un gran avance en la etapa de educación de requisitos de la ingeniería de software, es importante resaltar que este trabajo lo realiza utilizando lenguaje controlado, el cual minimiza la expresividad del interesado, además, para el uso de este lenguaje debe mediar una capacitación previa entre interesado y analista.

4. MÉTODO PROPUESTO

Con el fin de llevar a cabo la propuesta, se tomará como referencia el diagrama causa-efecto presentado por [8], el diagrama de Objetivo de KAOS presentado por [9] y el universo del discurso presentado en lenguaje natural.

De igual manera, se emplearán las entrevistas para el manejo del universo del discurso con el usuario, para lo cual se establece lo siguiente: (i) construcción de plantilla para realizar las entrevistas; (ii) conceptos lingüísticos de los elementos que conforman las reglas gramaticales; y (iii) Definición de las reglas gramaticales para la obtención del diagrama causa efecto.

Para la construcción de la plantilla (tabla 1), se partió de la importancia del lenguaje natural como base para la toma de requisitos, pues, se depende de éste para realizar una buena entrevista que permita la elaboración del diagrama causa efecto y posteriormente el diagrama de objetivo de KAOS:

4.1 Definición de conceptos gramaticales

Para la construcción de las reglas gramaticales se tomaron los siguientes conceptos del [24], Diccionario de la Real Academia de la Lengua Española:

“*Sujeto*: función oracional desempeñada por un sustantivo, un pronombre o un sintagma nominal en concordancia obligada de persona y de número con el verbo. Pueden desempeñarla también cualquier sintagma o proposición sustantivados, con concordancia verbal obligada de número en tercera persona”.

“*Adverbio*: palabra invariable cuya función consiste en complementar la significación del verbo, de un adjetivo, de otro adverbio y de ciertas secuencias. Hay adverbios de lugar, como aquí, delante, lejos; de tiempo, como hoy, mientras, nunca; de modo, como bien, despacio, fácilmente; de cantidad o grado, como, bastante, mucho, muy; de orden, como primeramente; de afirmación, como sí; de negación, como no; de duda o dubitativos, como acaso; de adición, como además, incluso, también; de exclusión, como exclusive, salvo, tampoco. Algunos pertenecen a varias clases”.

“*Gerundio*: Forma invariable no personal del verbo, cuya terminación regular, en español, es -ando en los verbos de la primera conjugación, -iendo o -

yendo en los de la segunda y tercera. Amando, temiendo, partiendo. Suele denotar acción o estado durativos. Estoy leyendo. Seguiré trabajando. Tiene más generalmente carácter adverbial, y puede expresar modo, condición, tiempo, motivo, concesión y otras circunstancias. Vino corriendo. Hablando se entiende la gente. Se emplea a veces en construcciones absolutas”.

“*Infinitivo*: Forma no personal del verbo, que en español lleva las terminaciones -ar, -er, -ir. En español y otras lenguas identifica o da nombre al verbo”.

“*Complemento*: que expresa circunstancias de la acción verbal, como lugar, tiempo, modo, instrumento”.

Perífrasis con gerundio: [25], define la principal característica de las perífrasis con gerundio es que tiene un carácter durativo que viene del gerundio, así mismo, señala algunas condiciones necesarias para hablar de perífrasis de gerundio una de ellas es que el verbo auxiliar debe estar gramaticalizado, otra es que el gerundio debe actuar sólo como verbo dejando las características de adverbio y la última condición es que el verbo auxiliar forma junto con el gerundio un núcleo. Algunos ejemplos de perífrasis con gerundio son: andar + gerundio, estar + gerundio, ir + gerundio.

Perífrasis con Infinitivo: [25], define las perífrasis con infinitivo de carácter progresivo y orientado al futuro. Además expresan combinaciones de valores aspectuales, modales y temporales; de igual manera para unir la forma personal con el infinitivo existen dos maneras: directamente (con o sin nexos) de tal manera que el verbo en forma personal es quien define si lo requiere o no (ej. tener que estudiar) o de las preposiciones a, de, por, para. Para que exista una perífrasis verbal de infinitivo, el verbo auxiliar tiene que estar gramaticalizado, eso quiere decir que el verbo auxiliar conserva su significado original; también se debe tener en cuenta que el infinitivo funcione como verbo y no sea confundido como nombre, pronombre o por un sintagma nominal. Dentro de las perífrasis con infinitivo existen las de modal de obligación donde su verbo auxiliar aparece gramaticalizado.

Tabla 1. Plantilla de entrevistas. Fuente: elaboración propia

ENTREVISTA ANALISTA - USUARIO	
Objetivo: esta entrevista tiene como propósito lograr que el interesado exprese con sus propias palabras las necesidades y/o problemas que tiene la organización, de tal forma, que permita entregar los primeros insumos para capturar los requisitos.	
Nombre del entrevistado (nombre completo)	
Cargo (cargo que desempeña el usuario)	
Nombre de la empresa (nombre completo de la empresa)	
Misión (colocar la misión de la empresa)	
Visión (colocar la visión de la empresa)	
Actividad económica (a que se dedica la empresa)	
Área del problema (describa el área en el que se presenta el problema)	
Funciones del área del problema (colocar cada una de las actividades que se desarrollan en el área del problema)	
Actores (las personas que intervienen en el área del problema con sus respectivas funciones)	
Problema (describir con claridad cada uno de los problemas que se están presentando en el área)	
Terceros (coloque si estos problemas afectan otras áreas de la empresa)	

4.2 Construcción de las reglas gramaticales para obtener el diagrama causa-efecto a partir del lenguaje natural

Teniendo en cuenta las definiciones presentadas en el numeral 4.2, se crearon las siguientes reglas gramaticales para la construcción del diagrama causa efecto, permitiéndole al analista mejorar su eficiencia al momento de la toma de requisitos.

La regla número uno (R1) para definir el problema principal es la siguiente:

R1: Sujeto + adverbio de negación + perífrasis durativa + gerundio + verbo en infinitivo + complemento.

Si en la entrevista se presenta la R1, entonces se puede obtener el problema principal para el diagrama causa efecto. Un ejemplo para esta regla

es el siguiente: *“La pizzería no está logrando ganar suficiente dinero”*.

En la oración anterior, se puede visualizar algunos de los conceptos de la real academia, como se muestran a continuación:

(i) Sujeto: La pizzería; (ii) Adverbio de Negación: No; (iii) Perífrasis Durativa: Está; (iv) Gerundio: Logrando; (v) Verbo en Infinitivo: Ganar; (vi) Complemento: Suficiente dinero.

La regla número dos (R2) para definir las causas es la siguiente:

R2: Sujeto + adverbio de negación + perífrasis durativas + gerundio + complemento

Si en la entrevista se presenta la R2, entonces se pueden obtener las causas para el diagrama causa-efecto. Un ejemplo para esta regla es el siguiente: *“El repartidor no está entregando el pedido al cliente en el tiempo establecido”*.

En la oración anterior, se puede visualizar algunos de los conceptos de la real academia, como se muestran a continuación:

Sujeto: El Repartidor; (ii) Adverbio de Negación: No; (iii) Perífrasis Durativa: Está; (iv) Gerundio: Entregando; (v) Complemento: El pedido al cliente en el tiempo establecido.

4.3 Construcción de las reglas gramaticales para obtener el diagrama de objetivos de KAOS a partir del lenguaje natural

Teniendo en cuenta las definiciones presentas en el numeral 4.2, se crearon las siguientes reglas gramaticales para la construcción del diagrama de objetivos de KAOS que permitirán articular el diagrama de objetivos de KAOS con el diagrama causa-efecto, en donde el problema principal se convertirá como objetivo de alto nivel y las causas en los requisitos.

RO1: Sujeto + perífrasis de obligación + infinitivo + complemento

La regla número uno (RO1) permite caracterizar desde el lenguaje natural el objetivo de alto nivel del diagrama de objetivo de KAOS:

Una vez hallado el problema principal del diagrama causa-efecto, éste, se organiza de manera positiva para caracterizar el objetivo de alto nivel del diagrama de objetivos de KAOS. Un ejemplo para

esta regla es el siguiente: *“La pizzería tiene que incrementar sus ganancias”*.

En la oración anterior, se puede visualizar algunos de los conceptos de la real academia, como se muestran a continuación:

(i) Sujeto: La pizzería; (ii) Perífrasis de obligación: Tiene que; (iii) Verbo en Infinitivo: Incrementar; (iv) Complemento: Sus ganancias.

RO2: Sujeto + verbo + complemento

La regla número dos (RO2) permite caracterizar desde el lenguaje natural el requisito y el actor del diagrama de objetivo de KAOS.

Una vez hallada la causa del diagrama causa-efecto, ésta se organiza de manera positiva para caracterizar los requisitos y los actores que cumplen con el requisito del diagrama de objetivos de KAOS. Para este caso, el sujeto será el actor del diagrama de objetivos y el requisito se compone del verbo más el complemento. Un ejemplo para esta regla es el siguiente: *“El repartidor entrega el pedido al cliente a tiempo”*.

En la oración anterior, se puede visualizar algunos de los conceptos de la real academia, como se muestran a continuación:

(i) Sujeto: El repartidor; (ii) Verbo: Entrega; (iii) Complemento: El pedido al cliente a tiempo.

5. RESULTADOS

El caso de estudio que ha sido anteriormente trabajado por otros autores en lenguaje controlado, ahora se trabajará tomando en cuenta las reglas gramaticales construidas y se plasmará en la plantilla presentada para la entrevista del analista con el usuario.

[10], plantea el caso de estudio del controlador avanzado y automático de los trenes de la siguiente manera: “el problema, se dirige 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 segmento 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), éste lo golpearía”.

A continuación se presentará el mismo caso de estudio, a través del método propuesto en este artículo. El método inicia a partir de la entrevista (en lenguaje natural o universo del discurso) utilizando la plantilla propuesta en este mismo trabajo y aplicando las reglas planteadas para la construcción del diagrama causa-efecto y el diagrama de objetivo de KAOS. Ver Tabla 2.

Reglas gramaticales del diagrama causa-efecto

R1. El sistema de control no está logrando garantizar un buen desplazamiento de los trenes.

R2. El tren no está controlando la velocidad de desplazamiento.

R2. El sistema de control no está garantizando que la vía sea la adecuada para el desplazamiento.

R2. El sistema de control no está logrando el control de la aceleración para los trenes.

R2. El sistema de control no está controlando la distancia entre cada recorrido de los trenes.

R2. El sistema de control no está enviando la señal a tiempo a cada tren.

En la figura 4 se logra observar el diagrama causa efecto construido a partir de las reglas gramaticales planteadas en la plantilla de entrevista analista usuario utilizando lenguaje natural.

Las reglas gramaticales para obtener el diagrama de objetivo de KAOS se presentan a continuación:

RO1. El sistema de control tiene que lograr el desplazamiento de los trenes.

RO2. El tren regula su velocidad de desplazamiento.

RO2. El sistema de control mantiene las vías en buen estado.

RO2. El sistema de control modera la aceleración de los trenes.

RO2. El sistema de control vigila la distancia entre los trenes.

RO2. El sistema de control manda la señal a tiempo al tren.

Partiendo de las reglas gramaticales construidas para la articulación del diagrama causa efecto con el diagrama de objetivos de KAOS se logra obtener el diagrama de objetivos de KAOS que se ilustra en la figura 5.

Tabla 2. Plantilla de entrevistas Controlador avanzado y automático de trenes. Fuente: elaboración propia.

ENTREVISTA ANALISTA - USUARIO	
Objetivo: esta entrevista tiene como objeto que el interesado relate con sus propias palabras las necesidades y/o problemas que presenta actualmente la organización, puesto que, ello permitirá capturar el problema y sus causas, además, permitirá capturar requisitos de una posible solución a través de una pieza de software.	
Nombre del entrevistado (nombre completo)	Actor responsable Controlador avanzado y automático de trenes
Cargo (cargo que desempeña el usuario)	Administrador Controlador avanzado y automático de trenes
Nombre de la empresa (nombre completo de la empresa)	Controlador avanzado de los trenes (San Francisco BART)
Misión (colocar la misión de la empresa)	Garantizar el desplazamiento de los trenes de servicio público de una manera ágil y segura.
Visión (colocar la visión de la empresa)	N/A
Actividad económica (a que se dedica la)	Administración de transporte público férreo
Área del problema (describa el área en el que se presenta el problema)	El área donde se presenta el problema es el centro de control en el cual se está desarrollando un sistema de control encargado de la aceleración y velocidad para obtener que los trenes corran rápida y suavemente
Funciones del área del problema (colocar cada una de las actividades que se desarrollan en el área del problema)	<ul style="list-style-type: none"> • 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 segmento 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), éste lo golpearía.
Actores (las personas que intervienen en el área del problema con sus respectivas funciones)	En este caso se presenta un solo actor que es el sistema, el cual es el encargado de verificar que los trenes estén funcionando correctamente.
Problema (describir con claridad cada uno de los problemas que se están presentando en el área)	Los problemas que se presentan son los siguientes: <ol style="list-style-type: none"> 1. El sistema de control no está logrando garantizar un buen desplazamiento de los trenes. 2. El tren no está controlando la velocidad de desplazamiento 3. El sistema de control no está garantizando que la vía sea la adecuada para el desplazamiento 4. El sistema de control no está logrando el control de la aceleración para los trenes 5. El sistema de control no está controlando la distancia entre cada recorrido de los trenes 6. El sistema de control no está enviando la señal a tiempo a cada tren.
Necesidades	
Terceros (coloque si estos problemas afectan otras áreas de la empresa)	

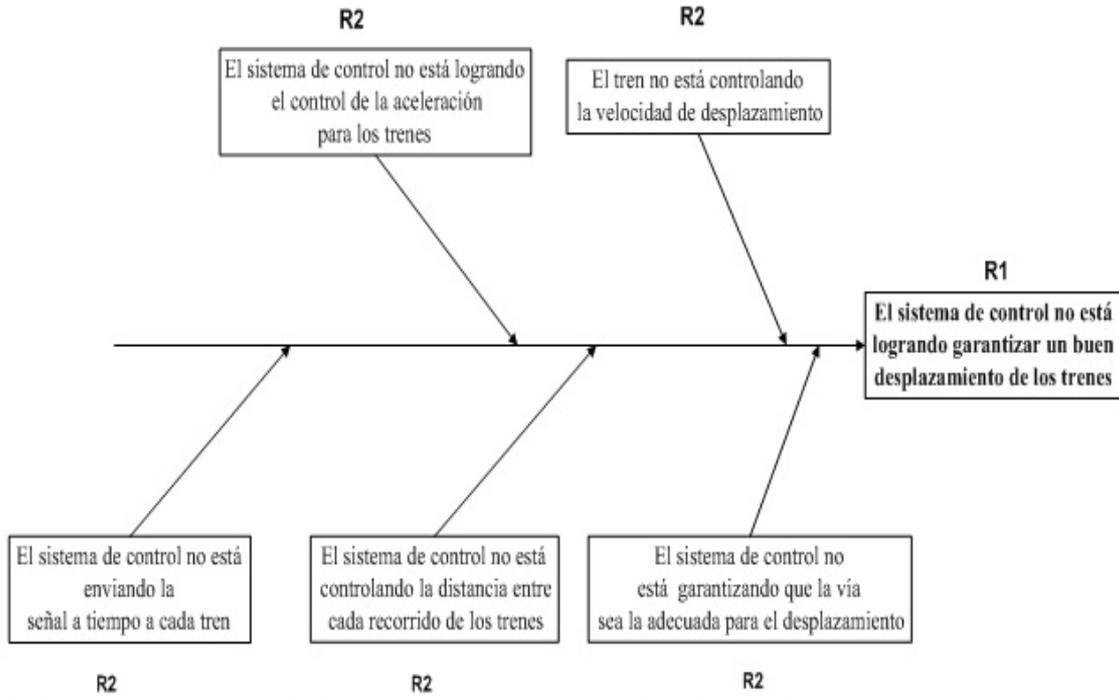


Figura 4. Diagrama causa-efecto Controlador avanzado y automático de trenes. Fuente: elaboración propia.

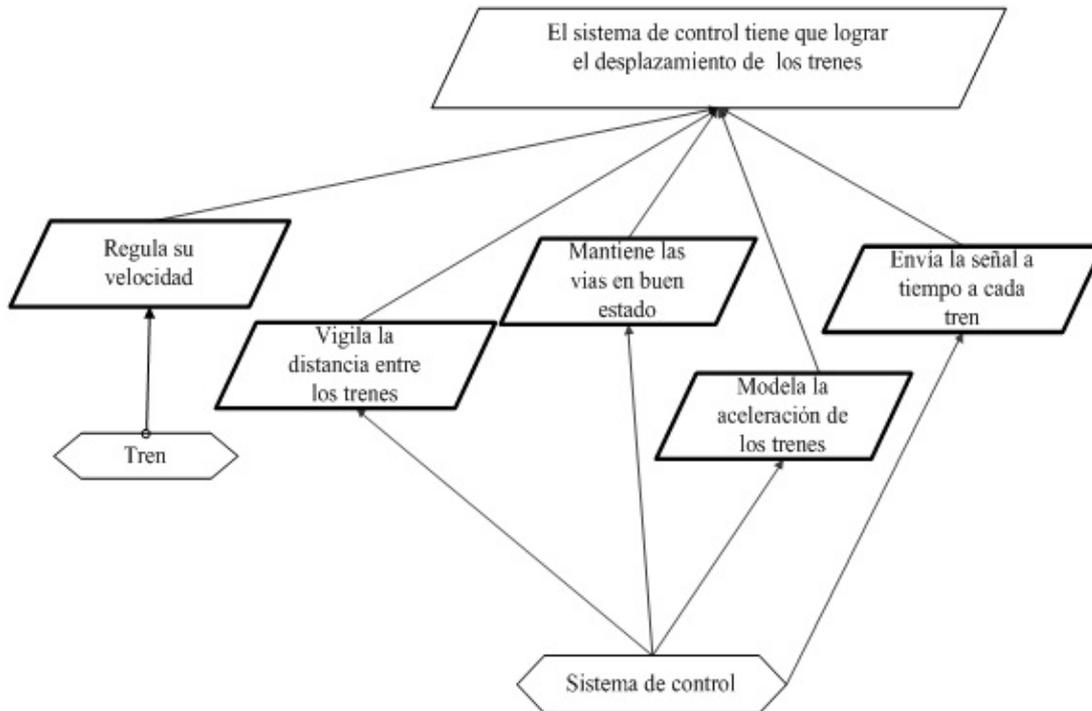


Figura 5. Diagrama de Objetivos de KAOS. Caso controlador avanzado y automático de trenes. Fuente: elaboración propia.

6. TRABAJO FUTURO

Los siguientes temas pueden dar continuidad al trabajo futuro de esta tesis:

- Automatizar el diagrama de objetivo de KAOS y el de causa-efecto a partir de las reglas gramaticales propuestas en este artículo.
- Definir reglas gramaticales que permitan caracterizar a partir del lenguaje natural cada una de los elementos del diagrama de objetivos como propiedad del dominio, expectativa, relación de responsabilidad, conflicto, AND y OR.
- Definir casos de estudio que nos permitan analizar los resultados aplicando las reglas propuestas en este artículo y la metodología utilizada en la actualidad.
- Caracterizar reglas para identificar los requisitos no funcionales

7. CONCLUSIONES

Para brindar una solución a la problemática que se da en las primeras etapas del ciclo de vida del software, en este trabajo se define un método para la educación de requisitos de software; para esto se diseñó una plantilla de entrevista en la cual se capturan los primeros datos a partir del lenguaje natural que darán inicio a la fases de análisis del software a construir.

Igualmente se definieron las reglas gramaticales que permiten a partir de la plantilla de entrevistas identificar la causa principal y las causas menores para la construcción del diagrama causa-efecto, lo cual permitirá al analista minimizar las ambigüedades y ser más certero a la hora de encontrar el problema principal a solucionar y las causas que lo están afectando.

Una vez definido el diagrama causa-efecto se construyeron unas nuevas reglas que permiten crear el diagrama de objetivo de KAOS logrando así la articulación entre estos dos diagramas. Finalmente se realiza un caso de estudio en el cual se muestra los resultados obtenidos a partir de la plantilla de entrevista y reglas gramaticales propuestas donde se evidencia la articulación del diagrama de objetivo de KAOS y el diagrama causa-efecto.

8. REFERENCIAS

- [1] ESTÉVEZ, E. Y FILLOTTRANI, P. (2008). Propuesta de Creación de la Carrera Ingeniería en Sistemas de Software. LISSI Departamento de Ciencias e Ingeniería de la Computación Universidad Nacional del Sur, 8 de abril de 2008.
- [2] SOMMERVILLE, I. (2005). Ingeniería del Software. Madrid. Séptima Edición.
- [3] ALARCÓN, A., SANDOVAL, V (2008). Herramientas Case para Ingeniería de Requisitos. Grupo de participación libre en investigación y desarrollo GPL@I+D. ISSN 1657-463X.
- [4] GONZÁLEZ, L. URREGO, G. (2008). Modelo de Requisitos para Sistemas Embebidos. Revista de Ingenierías Universidad de Medellín. Vol 7. Numero 13. pp 111-127
- [5] GÓMEZ, J. URREGO, G. GONZALEZ, L. (2009). REASEM: Herramienta para la Gestión de Requisitos. Revista avances en sistemas e informática. Vol 6. No 2.D.
- [5] A. de la Villa & A. Gómez. “Estimadores de estado generalizados de sistemas eléctricos de potencia”. Ingeniería Energética y Medioambiental, año XXXI, No. 186, pp. 64-69, jul.-ago. 2005.
- [6] ZAPATA, C. Y CARMONA, N. (2008). Un Modelo de Dialogo para la Educación de Requisitos de Software. Revista de la facultad de minas-Universidad nacional de Colombia – Sede Medellín DYNA. Nro. 164, pp. 209-219.
- [7] ARANGO, F. Y ZAPATA, C. (2006). UN-MÉTODO para la elicitación de requisitos de software. Escuela de Sistemas de la Universidad Nacional de Colombia, Medellín, Colombia. 113 p.
- [8] VARGAS, F. (2010). Método para establecer la consistencia de los problemas en el diagrama causa efecto con el diagrama de objetivos de KAOS. Tesis de maestría dirigida por Ph D. Carlos Mario Zapata Jaramillo. Universidad Nacional de Colombia. Enero 2010.
- [9] LAMSWEERDE AXEL VAN. (2000). Requirements Engineering in the Year 2000: A Research Perspective, 22nd International Conference on Software Engineering.
- [10] LEZCANO, L. (2007). Elaboración Semiautomática del Diagrama de Objetivos. Universidad Nacional de Colombia. Facultad de Minas-Escuela de Sistemas. Maestría en Ingeniería de Sistemas, Área de Ingeniería de Software Medellín, Colombia. Noviembre de 2007.
- [11] LEZCANO, L. (2012). The application of and unresolved problems regarding the use of

objectives in software engineering. Revista Ingeniería E Investigación ISSN: 0120-5609. Universidad Nacional de Colombia v.32 fasc.2 p.63 - 67.

[12] ÁLVAREZ, D., CABEZUELO, E. y SAN ROMÁN, A. (2005-2006). "Sistema de Asistencia Interactiva en Lenguaje Natural", Proyecto de Sistemas Informáticos Curso 2005-2006, Facultad de Informática Universidad Complutense de Madrid.

[13] CONTRERAS, H (2001). Procesamiento del lenguaje natural basado en una "gramática de estilos" para el idioma español. Universidad de los Andes, Bogotá, Colombia 54 p.

[14] GELBUKH, A (2010). Procesamiento del lenguaje natural y sus aplicaciones. Korpus Sapiens. Sociedad Mexicana de inteligencia artificial. Vol I.

[15] ROMERO, E., DIAZ, J. (2010). El Uso del Diagrama Causa – Efecto en el Análisis de Casos. Revista Latinoamericana de Estudios Educativos. Vol. XL. No 3-4. Pp.127-142.

[16] GRIMOLIZZI O., MAYER H., GOMEZ E., DE LLOMPARTE F., (2005). Empleo de Diagramas de causa efecto como Etapa Preliminar al Análisis para la Optimización del Tratamiento del Dolor Agudo Postoperatorio. Revista Argentina de Anestesiología, 63, 4: 195-201.

[17] FERNÁNDEZ, G. (2010). Propuesta de Modelo para la evaluación de la Sostenibilidad en la Dirección Integrada de Proyectos de Ingeniería Civil. Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos. Madrid.

[18] ZAPATA C, ARANGO F, (2004).Alineación Entre Metas Organizacionales Y Elicitación De Requisitos Del Software. Revista de la facultad de minas- Universidad nacional de Colombia – Sede Medellín DYNA. Vol 71 numero 143. pp 101-110.

[19] ANTÓN, A. (1996). Goal-Based Requirements Analysis. Proceedings of the Second IEEE International Conference on Requirements Engineering. Colorado Springs, USA, pp. 136-144.

[20] ANTÓN, A. (1997). Goal Identification and Refinement in the Specification of Software-Based Information Systems, PhD Thesis, Georgia Institute of Technology, Atlanta, USA.

[21] ANTÓN, A. (1998). The Use of Goals to surface Requirements for Evolving System. Software Engineering. Proceedings of the 1998 (20th) International Conference. Kyoto, Japan, pp. 157-166.

[22] ZAPATA C., VILLEGAS S., Y ARANGO F. (2006). Reglas de Consistencia entre Modelos de Requisitos de Un-Método. Universidad Eafit, enero-

marzo, año / vol. 42, número 141 Universidad Eafit Medellín, Colombia pp. 40-59.

[23] ZAPATA C., LEZCANO L. Y TAMAYO P. (2007). Validación del Método para la Obtención del Diagrama de Objetivos Desde Esquemas Preconceptuales. Revista EIA, ISSN 1794-1237 Número 8, p. 21-35. Diciembre 2007. Escuela de Ingeniería de Antioquia, Medellín (Colombia).

[24] Diccionario de la Real Academia de la Lengua Española 2012. <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>

[25] GEHOVÁ, K (2008). Perífrasis Verbales en la Prensa Española.