

Reglas Sintáctico-semánticas para Relacionar los Objetivos Organizacionales y los Problemas en el Contexto de la Educación Temprana de Requisitos de Software

Carlos Mario Zapata Jaramillo
Facultad de Minas
Universidad Nacional de Colombia
Medellín, Colombia
cmzapata@unal.edu.co

Fabio Alberto Vargas Agudelo
Facultad de Ingeniería
Tecnológico de Antioquia
Medellín, Colombia
fvargas@tdea.edu.co

Resumen—Los procesos de ingeniería de software de alta calidad requieren la educación temprana de requisitos funcionales y no funcionales. Este proceso lo llevan a cabo los analistas y los interesados, tratando de solucionar los problemas de información de la organización y contrastándolos con el contexto del negocio, representado en sus objetivos organizacionales. Este proceso se suele realizar de forma manual y, si bien existen algunos trabajos que establecen una relación entre los problemas y los objetivos, se basan en la negación total de unos para obtener los otros. Esto conduce al desarrollo de aplicaciones de software no pertinentes para la organización, que no resuelven sus problemas prioritarios o que no se alinean con sus objetivos organizacionales. Por estas razones, en este artículo se propone una relación entre los problemas a solucionar y los objetivos organizacionales, empleando un conjunto de reglas sintáctico-semánticas que validen dicha relación y que se reflejen en requisitos consistentes y contextualizados con la organización. Estas reglas se validan con un caso de estudio.

Palabras clave—Objetivos Organizacionales, problemas, educación temprana de requisitos, reglas sintácticas, reglas semánticas.

I. INTRODUCCIÓN

La ingeniería de software (IS) viene evolucionando en su proceso de automatización, permitiendo a los analistas e interesados mejorar en muchos aspectos los métodos en los cuales participan en procura de lograr desarrollos de software de alta calidad. Cuando se desarrolla una aplicación de software en cualquier plataforma y para cualquier entorno, es de vital importancia reconocer y establecer condiciones que garanticen la pertinencia, calidad, seguridad, eficiencia y rendimiento del aplicativo de software que se implementa [1]. Por tal razón, es importante llevar a cabo las etapas del desarrollo de software (definición, análisis, diseño, desarrollo, pruebas, implementación y mantenimiento), que suministren unas pautas que conlleven al buen desarrollo del aplicativo [2]. La IS está generando propuestas de métodos, artefactos y estrategias metodológicas que buscan darle al desarrollo de software orden, estandarización y calidad. Con esto, se busca que las aplicaciones que se desarrollen se adapten a los diferentes interesados, pasando por personas u organizaciones que las requieran y teniendo en cuenta sus necesidades,

expectativas y, muy especialmente, tomando en cuenta los objetivos de la organización, a fin de garantizar su cumplimiento [3]. La IS busca, en su fase de educación temprana de requisitos, un reconocimiento muy profundo de los problemas que se presentan en la organización y de los objetivos que pretenden alcanzar los actores involucrados en los diferentes procesos, a fin de proponer soluciones (aplicaciones de software) y tomar decisiones que se ligen de forma directa con los objetivos organizacionales [4]. De esta manera, se busca la especificación consistente de requisitos funcionales y no funcionales.

Rebollar *et al.* [5] reconocen la importancia de las técnicas de ingeniería de requisitos temprana, conocidas como técnicas de modelado organizacional, ya que es la etapa fundamental para la construcción de software de alta calidad que dé respuesta a las necesidades y expectativas de los interesados y que tenga muy en cuenta los objetivos de la organización, garantizando un software contextualizado y pertinente [6]. Se reconoce que el contexto puede en un momento dado influir en los objetivos de los interesados y, por ende, en la forma de conseguirlos [6]. La captura de requisitos es un proceso manual que lleva a cabo el analista con base en su experiencia e interpretación. En esta etapa, la definición de los problemas a solucionar y su relación con los objetivos de la organización se realizan sin seguir unas pautas previas que garanticen la consistencia y, en muchos casos, esto trae consigo problemas posteriores en el ciclo de vida del desarrollo de software [7]. Existen algunos trabajos que plantean relaciones de consistencia entre objetivos y problemas, pero aún se quedan cortos, pues sólo establecen las relaciones con base en la negación total de los objetivos para obtener los problemas [2].

En este artículo se propone un conjunto de reglas sintácticas y semánticas para establecer el vínculo entre los objetivos y los problemas en el contexto de la educación temprana de requisitos. Para ello, se realiza una revisión y análisis de las metodologías para la educación temprana de requisitos de software que utilizan directamente objetivos y problemas para la especificación de los mismos, con el fin de verificar cómo estructuran sus objetivos y problemas y, además, cómo se relacionan entre ellos, para mejorar la consistencia y evitar futuros problemas en el desarrollo. Se busca determinar cuáles problemas son prioritarios para la

organización y, a partir de ellos, lograr una especificación consistente de requisitos de software funcionales y no funcionales.

El artículo se estructura de la siguiente forma: en la Sección 2 se realiza una revisión y análisis de las metodologías para representar objetivos y problemas en el proceso de educación de requisitos y en el análisis organizacional; en la Sección 3 se define la metodología establecida para proponer las reglas; en la Sección 4 se presenta una propuesta sintáctica-semántica para relacionar los objetivos y los problemas; en la Sección 5 se presenta un caso de estudio basado en un diagrama de objetivos y en un diagrama causa-efecto; finalmente, se presentan las conclusiones y el trabajo futuro

II. ANTECEDENTES

A. Representación de objetivos y problemas en los procesos de educación temprana de requisitos

Los objetivos constituyen los referentes principales para la toma de decisiones a nivel organizacional, ya que son los ejes que rigen cualquier proceso de negocios. Por tal motivo, deben poseer una coherencia en su representación y expresión que garantice a interesados y analistas su interpretación y relación con otros componentes básicos de algún proceso que se inicie en la organización. A nivel de desarrollo de software, es necesario facilitar la representación de los objetivos en los diferentes diagramas y modelos utilizados en los procesos de educación temprana de requisitos. La representación de objetivos y problemas en el proceso de educación de requisitos se realiza con algunas metodologías como KAOS e I*.

KAOS (*Knowledge Acquisition autOmated Specification*) [8] plantea la representación de un árbol de objetivos, el cual se enfoca en realizar un proceso de análisis formal de los requisitos. 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 presentarlos en objetivos más elementales que los subrogan y así sucesivamente hasta llegar a objetivos que se consideren elementales o atómicos o hasta que se alcancen expectativas, requisitos o propiedades del dominio. Con el diagrama de objetivos se busca señalar cuáles de los objetivos subrogados satisfacen actualmente los procesos del área. Se debe notar que la incapacidad de dar satisfacción a los objetivos generales se asocia con la incapacidad de dar satisfacción a los objetivos subrogados. El diagrama de objetivos se realiza no sólo para el área problemática, sino para la organización que la rodea, porque se requiere que la aplicación de software del área se contextualice y se pueda determinar su influencia en la organización, evaluando la viabilidad de cumplimiento de los objetivos subrogados pertenecientes a esa área. El analista construye de manera subjetiva el diagrama, garantizando manualmente la consistencia entre sus elementos. Además, no se plantean estructuras claras para definir objetivos y, en algunos casos, los objetivos planteados dan mayor cuenta de operaciones y no realmente de objetivos. Zapata y Vargas [7] anotan que la estructura y definición de objetivos y requisitos dentro del diagrama de objetivos de KAOS es una tarea del analista, quien lo define de acuerdo con la interpretación y el conocimiento adquirido del área y de la organización, con base en las reuniones con los interesados y en la exploración documental realizada.

I* [9] es un lenguaje orientado a objetivos que incluye nodos que representan actores, objetivos, tareas y recursos,

además de un conjunto de relaciones entre ellos. Es un enfoque que utiliza la idea de *softgoal*. La principal particularidad del modelado de negocio sobre otros campos de la ingeniería de requisitos es la importancia de los agentes. Un agente se define como una entidad que existe en la organización, que tiene objetivos y que puede realizar tareas o utilizar recursos para alcanzar dichos objetivos o ayudar a otros agentes a alcanzar sus objetivos. I* tiene una alta dependencia del analista en la elaboración de los diagramas y tampoco existe una estructura sintáctico-semántica que ayude a estructurar adecuadamente los objetivos, para su fácil relación con otros componentes del sistema.

Zapata y Lezcano [10] plantean algunos problemas que se presentan en los diagramas de objetivos de KAOS e I*: es difícil automatizar el proceso porque los analistas suelen construir el diagrama de objetivos de forma subjetiva, tomando como base la información que suministran los interesados y se suele presentar una confusión entre los verbos que denotan objetivos y aquellos que expresan operaciones del dominio.

Rebollar *et al.* [5], Bresciani *et al.* [11] y Ali *et al.* [12, 13] plantean TROPOS como una metodología para el modelado organizacional, muy utilizada en los procesos de educación temprana de requisitos de software. Esta metodología permite capturar no sólo el qué o el cómo, sino también el porqué del desarrollo del software en la organización. Esta metodología, además, permite realizar una descripción detallada de las dependencias del sistema a desarrollar y logra una adecuada especificación de requisitos funcionales y no funcionales. TROPOS utiliza dos diagramas para la representación del ambiente organizacional, vitales en la educación temprana de requisitos que plantea la metodología: el diagrama de actores, el cual describe los actores, sus objetivos y las dependencias entre ellos, y el diagrama de objetivos, el cual muestra detalladamente la relación entre los objetivos y los actores [5]. Esta metodología continúa presentando los problemas que enuncian Zapata y Lezcano [10].

Ali *et al.* [6] plantean un modelo de Ingeniería de requisitos orientado hacia objetivos como extensión de TROPOS, donde se especifica que estos son una abstracción útil de las necesidades y expectativas de los interesados y que facilitan su representación.

Para Choi *et al.* [14] los requisitos en lenguaje natural se expresan mediante objetivos y escenarios, utilizando una estructura semántica para representarlos. Los objetivos, en general son oraciones que se representan mediante un verbo, un objeto conceptual o físico, una dirección de origen o destino y la forma o camino en que se van a lograr. Los escenarios se representan mediante una oración que contiene los siguientes componentes: un agente o actor, un verbo, un objeto conceptual o físico, una dirección de origen o destino y una forma o camino.

Zapata y Vargas [15] especifican reglas gramaticales para enunciar problemas, objetivos y la relación entre ellos, que se aplican en el proceso de educación de requisitos de software, así como en el análisis organizacional. En las Tablas I, II y III se muestran ejemplos de la aplicación de las reglas. Las abreviaturas empleadas en las tablas, son las siguientes: O=Oración, V=Verbo, Ad=Adjetivo, SN=Sintagma Nominal, Adv=Adverbio, S=Sustantivo, V1=Verbo, V2=Verbo, Ad=Adjetivo, SN1=Sintagma Nominal, SN2=Sintagma Nominal, Adv1=Adverbio, Adv2=Adverbio, C=Conjunción.

TABLA I. ESTRUCTURAS GRAMATICALES PARA ENUNCIAR PROBLEMAS [15]

Caso	Descripción	Restricciones	Ejemplos
1	O→V+Ad+S N	V→{en forma reflexiva impersonal o voz pasiva refleja, por ejemplo: "hay", "existe", "presenta"} Ad→{debe tener una connotación negativa; se sugieren palabras como "bajo", "poco", "malo", etc.}	Se presenta baja producción de materia prima en la fábrica de telares. Existen malas condiciones habitacionales en el conjunto residencial.

TABLA II. ESTRUCTURAS GRAMATICALES PARA ENUNCIAR OBJETIVOS [15]

Caso	Descripción	Restricciones	Ejemplos
1	O→V1+Ad+ SN	V1→{Verbo de logro} Ad→ {Debe tener connotación positiva. Por ejemplo: "Alta", "buena", "Adecuada"}	Lograr alta producción de materia prima en la fábrica de telares. Lograr buenas condiciones habitacionales en el conjunto residencial.

TABLA III. REGLAS DEFINIDAS PARA RELACIÓN ENTRE PROBLEMAS Y OBJETIVOS [15]

Caso	Descripción	Restricciones	Ejemplos
1	O→V+Ad1+ SN P→V1+Ad2+ SN	V→{Verbo de logro} V1 → {en forma reflexiva impersonal o voz pasiva refleja} SN→ {Se usa el mismo sintagma nominal tanto para el objetivo como para el problema.} Ad1 y Ad2→ {Los adjetivos deben poseer connotaciones contrarias.}	Objetivo: Lograr alta producción de materia prima en la fábrica de telares. Problema: Existe baja producción de materia prima en la fábrica de telares. Objetivo: Lograr buenas condiciones habitacionales en el conjunto residencial. Problema: Existen malas condiciones habitacionales en el conjunto residencial.

Eriksson y Penker [16] estructuran un modelo objetivo/problema que especifica los objetivos y subobjetivos de la organización e indica los problemas que se deben resolver a fin de alcanzar dichos objetivos. Este modelo se basa en una extensión del diagrama de objetos de UML. Este modelo no especifica estructuras para representar objetivos ni problemas y tampoco plantea estrategias para relacionarlos. Toda la tarea sigue siendo del analista basado en su experiencia y conocimiento.

B. Representación de objetivos y problemas en los procesos de educación temprana de requisitos

A nivel de análisis organizacional, Ortegón *et al.* [17] plantean que la metodología de Marco Lógico utiliza un árbol de objetivos y un árbol de problemas, en los cuales se representan los objetivos y las situaciones futuras a la que se desea llegar una vez se resuelvan los problemas plasmados en el árbol. Para la relación entre el árbol de objetivos y el árbol de problemas se tienen en cuenta algunas reglas como: los estados negativos encontrados en los problemas se convierten en estados positivos alcanzados; los problemas se reformulan convirtiéndolos en objetivos y el problema central detectado se

convierte también en un objetivo central del proceso. Todo este proceso lo lleva a cabo en forma manual el analista organizacional.

En la metodología Kepner-Tregoe [18] se implementa un procedimiento para la solución de problemas que facilita la toma de decisiones al interior de la organización. Para ello, se realiza una serie de etapas que incluyen el análisis de situaciones, el análisis de problemas, el análisis de objetivos de la decisión a tomar y el análisis de problemas potenciales a ocurrir después de la decisión tomada. La metodología exige que los objetivos describan los aspectos que se van a lograr en forma precisa y situarlos en un tiempo y un lugar definido, pero no plantea una estructura precisa para realizar esa descripción.

III. ANTECEDENTES

A. Fase de Exploración

En esta fase se realizó una revisión de literatura y un análisis de las metodologías de educación temprana de requisitos de software que utilizan problemas y objetivos, las formas de representación y la relación que puede existir entre problemas y objetivos a fin de identificar los avances y las falencias que existen en esta temática. Se realizó una exploración de estas mismas metodologías para determinar cómo llevan a cabo el proceso de captura de requisitos, procurando definir cuál es la tarea del analista y cómo él establece los problemas y los objetivos de la organización y su relación para determinar cuáles problemas requieren una solución prioritaria empleando una aplicación de software.

También, se analizaron los artículos en los cuales se especifican posibles estructuras sintáctico-semánticas para expresar objetivos y problemas que ayuden al analista de sistemas en la elaboración de los diferentes diagramas, profundizando en la oración y su forma de enunciarla.

B. Fase de Definición

Teniendo como base las estructuras de objetivos y problemas planteadas en los diferentes diagramas revisados, es decir KAOS [8], I* [9], TROPOS [5, 11, 12 y 13] y el modelo objetivo/problema [16] y además la fase de exploración realizada, se estableció un conjunto de reglas sintáctico-semánticas que permitieron la relación automática entre objetivos y problemas en los procesos de educación temprana de requisitos. Como base para la representación se emplearon los denominados esquemas preconceptuales [19] dada su cercanía con el lenguaje natural del interesado y la posibilidad de definir animaciones de la especificación en forma de esquemas preconceptuales ejecutables. La sintaxis básica de los esquemas preconceptuales se muestra en la Figura 1. En los conceptos se colocan los sustantivos o frases nominales del dominio; las notas contienen posibles valores de los conceptos; las relaciones estructurales se asocian con los verbos "ser" y "tener"; las relaciones dinámicas son actividades u operaciones del dominio; los condicionales son expresiones que pueden disparar relaciones dinámicas; las implicaciones son relaciones causa-efecto entre relaciones dinámicas o entre condicionales y relaciones dinámicas; las conexiones sirven para conectar conceptos con relaciones estructurales/dinámicas o viceversa; las conexiones concepto-nota sirven para ligar los conceptos con sus posibles valores. Para hacer ejecutable un esquema preconceptual simplemente hay que colocar los valores correspondientes en los conceptos hoja (aquellos que reciben una relación "tiene" y de los que no sale ninguna otra relación);

también, se representan los valores de los conceptos clase (los que no son conceptos hoja) por medio de tablas adicionales.

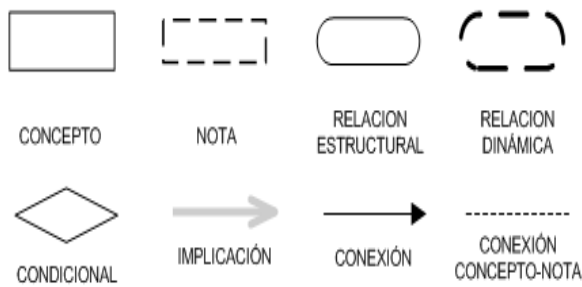


Fig. 1. Sintaxis básica de los esquemas preconceptuales.

C. Fase de Definición

En esta fase se desarrolló un caso de estudio donde se aplicaron estas reglas en la elaboración en la representación de las relaciones entre los objetivos y los problemas durante la etapa de educación temprana de requisitos de software.

IV. PROPUESTA

La propuesta que se describe en este artículo parte de los problemas encontrados en secciones anteriores, donde en las metodologías correspondientes a los procesos de educación temprana de requisitos de software no se especifican mecanismos que permitan relacionar de forma consistente y automática los objetivos organizacionales con los problemas que se desean solucionar con la aplicación de software, dejando toda la responsabilidad en la experiencia e intuición del analista y, en muchos casos, generando requisitos de software poco consistentes y descontextualizados con la organización. La propuesta incluye en conjunto de reglas sintáctico-semánticas que permiten relacionar un problema determinado con uno o varios objetivos organizacionales.

Las reglas se construyeron teniendo en cuenta el rol semántico y el rol sintáctico de cada frase que describe un objetivo y un problema. La relación entre un objetivo y un problema se establece con base en unas restricciones, cuya estructura se define mediante una fórmula. El esquema preconceptual que describe este proceso se muestra en la Figura 2.

V. CASO DE ESTUDIO

Para la ejemplificación del manejo del esquema preconceptual de la Figura 2, se tomó como base un ejemplo que presenta Zapata [20] con una estructura completa de un diagrama de objetivos y un diagrama causa-efecto. La restricción que se define con la fórmula de la Figura 3 se establece de la siguiente manera: si las frases que representan un objetivo y un problema comparten al menos un sustantivo y un verbo, el objetivo y el problema tienen una relación. Para el ejemplo, el problema P10 (no todos los álbumes se pueden acceder) y el objetivo O7 (asegurar que se pueda acceder a los álbumes) comparten el verbo acceder y el sustantivo álbumes, por lo cual el resultado de la relación se fija en verdadero. Nótese en la Figura 3 que las tablas que sirven para apoyar el proceso se ubican en la parte inferior izquierda, en tanto que

los diferentes valores se asocian con cada uno de los conceptos hoja.

VI. CONCLUSIONES

La educación temprana de requisitos de software, requiere, en su análisis organizacional, la identificación de problemas y objetivos y el establecimiento de una relación entre ellos, a fin de determinar qué problemas afectan el logro de un objetivo de la organización y, por ende, el buen desarrollo de la misma. Las relaciones de consistencia entre objetivos y problemas, pueden conducir a un mejor análisis de las soluciones problemáticas y, consecuentemente, al planteamiento de soluciones adecuadas y la definición consistente de requisitos funcionales y no funcionales alineados con la estrategia organizacional (sus objetivos) y que resuelvan las situaciones negativas (sus problemas).

La literatura especializada no plantea métodos automáticos que permitan relacionar objetivos y problemas pues, pese a que algunos utilizan técnicas de representación tanto de objetivos como de problemas (diagrama de objetivos de KAOS, diagrama causa-efecto, árbol de problemas y árbol de objetivos de la metodología de marco lógico), sigue siendo un trabajo que depende del analista, sin que medie ningún proceso de consistencia. Las relaciones sintáctico-semánticas que se propusieron para relacionar los problemas y objetivos facilitan la tarea del analista en los diferentes procesos de educación temprana de requisitos de software, estableciendo de forma automática la consistencia que existe entre objetivos y problemas.

La generación de soluciones automatizadas que apoyen a los analistas de sistemas en su proceso de educación temprana de requisitos de software, genera un mayor grado de confiabilidad en las soluciones de software planteadas, ya que éstas se alinearán con el contexto de la organización y serán pertinentes a las necesidades de la misma. Tener en cuenta en la solución los objetivos organizacionales garantiza una especificación más consistente de requisitos de software.

Como líneas de trabajo futuro, se encuentran las siguientes:

- La definición de diferentes restricciones que permitan establecer la relación entre objetivos y problemas, desde el punto de vista sintáctico y semántico.
- El estudio de otras relaciones entre problemas y objetivos que ya no se ligan con las mismas palabras. En estos casos, se deberían analizar relaciones de sinonimia y proximidad.
- El análisis de elementos pragmáticos que permitan establecer la relación entre objetivos y problemas. Por ejemplo, para el caso de estudio podría ser intuitivamente claro que el problema “no todos los álbumes se pueden acceder” ataca directamente el objetivo “incrementar los usuarios”, pero sus frases no tienen elementos comunes ni estructuras que permitan indicar la relación que puede existir entre el objetivo y el problema. Así, se requieren mecanismos adicionales que contribuyan a definir ese nexo.
- La construcción de una herramienta de análisis que esté en capacidad de determinar la relación entre los objetivos y problemas y luego trazar los diagramas de manera consistente.

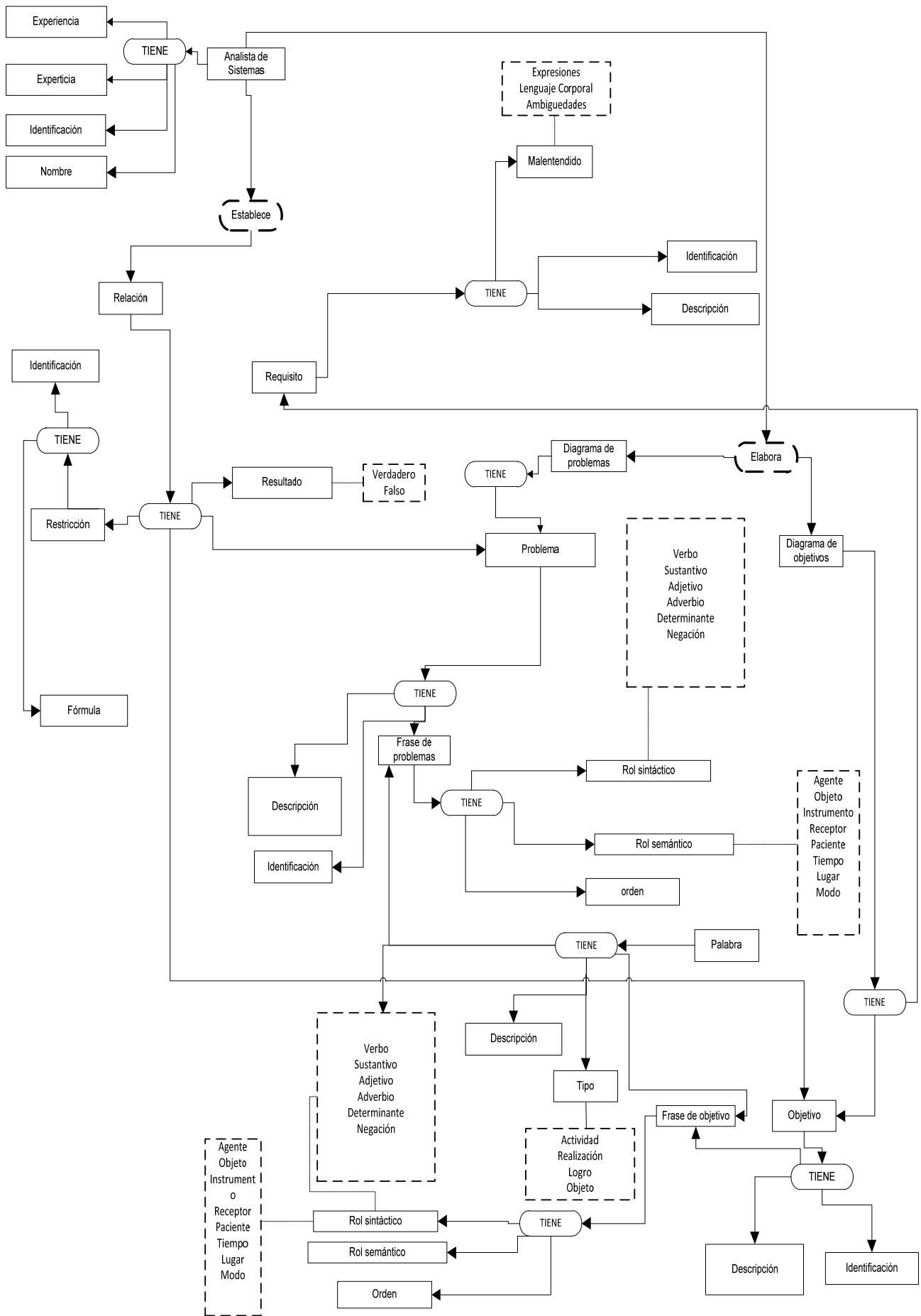


Fig. 2. Esquema preconceptual que representa la propuesta para relacionar objetivos y problemas.

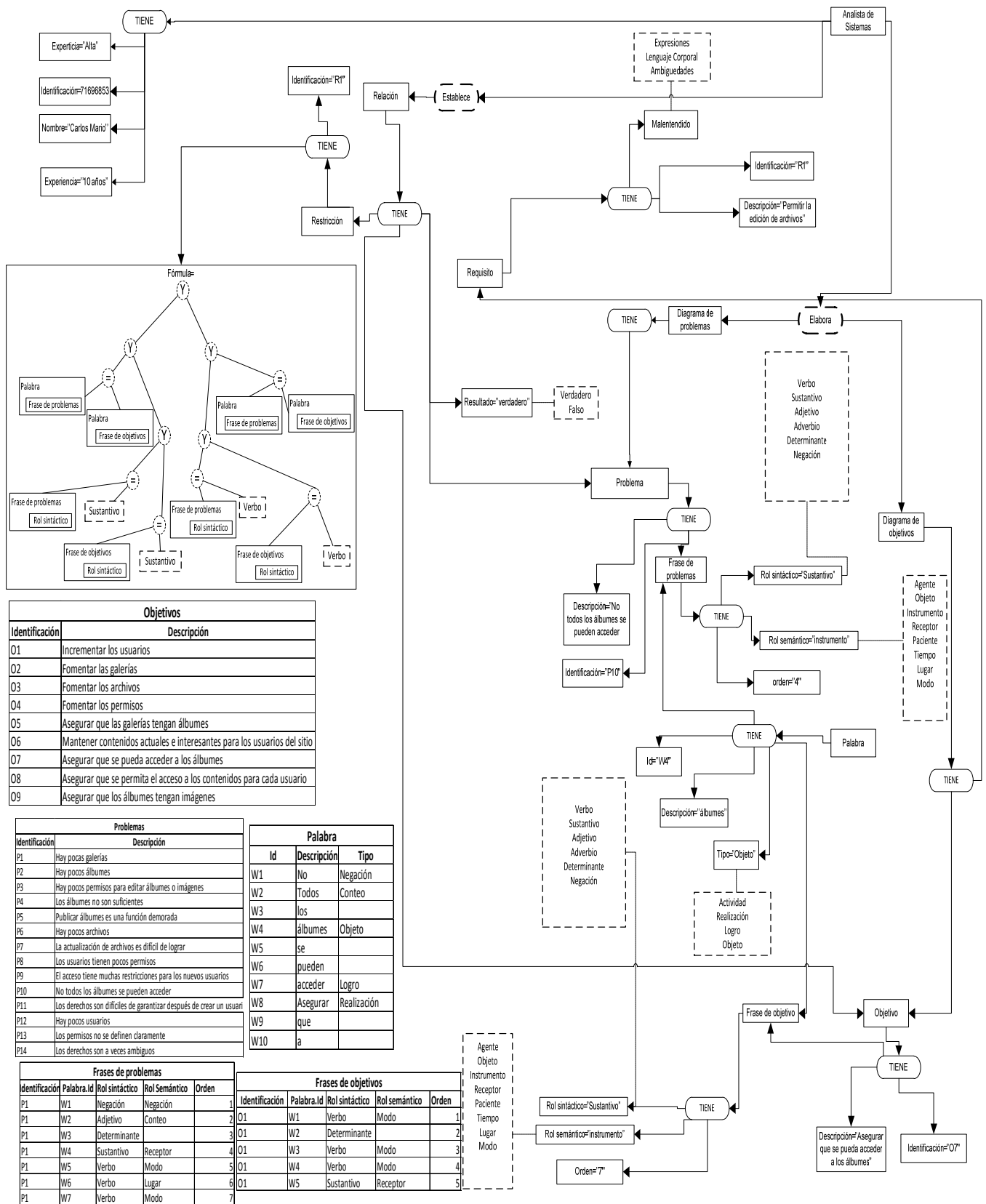


Fig. 3. Esquema preconceptual correspondiente al caso de estudio para el análisis de la relación entre objetivos y problemas.

REFERENCIAS

[1] R. Pressman, "Ingeniería del software. Un enfoque práctico", McGraw Hill, México, 2002.

[2] F. Vargas, "Método para establecer la consistencia de los problemas en el diagrama causa-efecto con el diagrama de

objetivos de KAOS", Tesis de maestría (Ingeniería de Sistemas). Universidad Nacional de Colombia, Medellín, 2010.

[3] M. G. Christel y K. C. Kang, "Issues in requirements elicitation", Technical Report, DTIC Document, Pittsburg, Estados Unidos, 1992.

- [4] C. M. Zapata y F. Arango, "Alineación entre metas organizacionales y elicitación de requisitos del software", *Revista Dyna*, vol. 143, 2004, pp. 101-110.
- [5] A. M. Rebollar, H. E. Esquivel, y L. A. G. Moreno, "una guía rápida de la metodología tropos", *Revista Gerencia Tecnológica Informática*, vol 7, nro. 19, 2008, pp. 67-77.
- [6] R. Ali, F. Dalpiaz, y P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis", *Requirements Engineering*, vol. 15, nro. 4, 2010, pp. 439-458.
- [7] C. Zapata y F. Vargas, "Una revisión de la literatura en consistencia entre problemas y objetivos en Ingeniería de Software y Gerencia Organizacional", *Revista Escuela de Ingeniería de Antioquia*, vol. 11, 2009, pp. 117-129.
- [8] A. Dardenne, A. Van Lamsweerde, y S. Fickas, "Goal-directed requirements acquisition", *Science of computer programming*, vol. 20, nro. 1, 1993, pp. 3-50.
- [9] E. Yu, "Modelling strategic relationships for process reengineering", *Social Modeling for Requirements Engineering*, vol. 11, 2011.
- [10] C. M. Zapata y L. A. Lezcano, "caracterización de los verbos usados en el diagrama de objetivos", *Revista Dyna*, vol. 76, nro. 158, 2009, pp. 219-228.
- [11] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, y J. Mylopoulos, "Tropos: An agent-oriented software development methodology", *Revista Autonomous Agents and Multi-Agent Systems*, vol. 8, nro. 3, 2004, pp.
- [12] R. Ali, F. Dalpiaz, y P. Giorgini, "Location-based software modeling and analysis: Tropos-based approach", *Journal Lecture Notes in Computer Science*, Vol 5231, 2008, pp 169-182.
- [13] R. Ali, F. Dalpiaz, y P. Giorgini, "Location-based variability for mobile information systems", *Technical Report in Advanced Information Systems Engineering*, 2008, pp. 575-578.
- [14] S. Choi, S. Park, y V. Sugumaran, "A rule-based approach for estimating software development cost using function point and goal and scenario based requirements", *Revista Expert Systems with Applications*, vol. 39, nro. 1, 2012, pp. 406-418.
- [15] C. Zapata y F. Vargas, "Innovation in project design and assessment: establishing linguistic relationships among goals and problems", *Revista Lámpsakos*, vol. 3, No. 6, 2011, pp. 46-55.
- [16] H. E. Eriksson y M. Penker, "Business modeling with UML: Business patterns at work". Mexico, 1999.
- [17] E. Ortégón, J. Pacheco y A. Prieto, "Metodología del marco lógico para la planificación, el seguimiento y la evaluación de proyectos y programas". Publicación de las Naciones Unidas, Santiago de Chile. 2005.
- [18] Ch. Kepner y B. Tregoe, "The New Rational Manager: An Updated Edition for a New World". Princeton Research Press, Princeton. 1997.
- [19] C. M. Zapata, A. Gelbukh y F. Arango, "Pre-conceptual schema: a conceptual-graph-like knowledge representation for requirements elicitation", *Lecture Notes in Computer Science*, vol. 4293, 2006, pp. 17-27.
- [20] Requirements elicitation", *Lecture Notes in Computer Science*, vol. 4293, 2006, pp. 17-27.



Carlos Mario Zapata Jaramillo. recibió su título de Ingeniero Civil en 1991, una Especialización en Gerencia de Sistemas Informáticos en 1999, una Maestría en Ingeniería de Sistemas en 2002 y un Doctorado en Ingeniería con énfasis en Sistemas en 2007. Todos los títulos los recibió en la Universidad Nacional de Colombia. Es Profesor Asociado del Departamento de Ciencias de la Computación y de la Decisión de la Universidad Nacional de Colombia, sede Medellín. Sus áreas de interés son: ingeniería de software, ingeniería de requisitos, lingüística computacional y estrategias didácticas para la enseñanza de la ingeniería.



Fabio Alberto Vargas Agudelo. Es Ingeniero de Sistemas por la Universidad Cooperativa de Colombia 1999, Especialista en Ingeniería de Software por la Universidad Nacional de Colombia 2003, y M.S. en Ingeniería de Sistemas por la Universidad Nacional de Colombia 2010. Es Profesor de tiempo completo categoría auxiliar y Líder del grupo de Investigación en Ingeniería de Software GIISTA en el Tecnológico de Antioquia (Institución Universitaria). Sus áreas de interés son: ingeniería de requisitos, pruebas de software y desarrollo de software. Actualmente es Candidato de Doctorado en Ingeniería de Sistemas por la Universidad Nacional de Colombia.