

# Método de consistencia en la relación entre problemas y objetivos para el proceso de elicitación de requisitos de software

## Consistency method in the relationship between problems and objectives for the process of software requirements elicitation

Fabio Vargas  
Magíster en Ingeniería de Sistemas  
fvargas@tdea.edu.co  
Docente  
Tecnológico de Antioquia, Medellín Colombia

*Recibido: 18 de febrero 2012  
Aprobado: 5 de abril 2012*

### Resumen

La captura de requisitos es un proceso manual que lleva a cabo el analista con base en su experiencia e interpretación. En este proceso, la definición de los problemas por solucionar y su relación con los objetivos de la organización se realizan sin seguir unas pautas que garanticen la consistencia. En muchos casos, esto trae consigo problemas posteriores en el ciclo de vida del software. Métodos de ingeniería de software utilizan el diagrama de objetivos de KAOS y el diagrama causa-efecto para representar objetivos y problemas, pero siguen siendo una tarea de interpretación del analista, sin tener en cuenta métodos de consistencia para su representación. El artículo aborda la generación de un método que establezca de forma automática y consistente la relación entre objetivos de la organización y los problemas que se detectan en el proceso de educación de requisitos, además de presentar un conjunto de estructuras lingüísticas para la representación de problemas y objetivos.

**Palabras clave:** Estructura de problemas, estructura de objetivos, relaciones de consistencia, estructuras gramaticales, educación de requisitos.

### Abstract

The capture of requirements is a manual process carried out by the analyst, based on his/her experience and interpretation. In this process, the definition of problems to be solved and its relationship with the goals of the organization is performed without following guidelines to ensure consistency, and, in many cases, this entails further problems in the software life cycle. Software engineering methods use KAOS goal diagram and cause-and-effect diagrams to represent goals and problems, but the task of interpretation continue to be for the analyst, regardless the consistent methods in its rendering. This paper addresses the generation of a method automatically and consistently establishing a relationship between the organization's goals and the problems detected in the requirements elicitation process, and delivers a set of linguistic structures to represent issues and goals.

**Keywords:** Problem structure, structure of goals, consistency relationships, grammatical structures, requirement elicitation.

## Introducción

Establecer una solución a una situación problemática en procesos de ingeniería de software requiere un conocimiento y un análisis muy profundo del área y de la organización en la cual se quiere implementar una aplicación de software que represente dicha solución. Para ello, es muy importante tener en cuenta aspectos vitales que garanticen la calidad de la solución planteada, entre los que se destacan la identificación y el conocimiento de los objetivos de la organización, así como la identificación y reconocimiento de los problemas (Zapata y Vargas, 2009). La identificación y la representación de estos elementos se llevan a cabo mediante ciertos diagramas o descripciones en lenguaje natural. Esa representación suele partir del análisis subjetivo, intuitivo y basado en la experiencia que realiza el analista responsable de tales procesos (Zapata y Vargas, 2009). La ingeniería de software viene generando propuestas de métodos, artefactos y estrategias metodológicas que buscan darle orden, estandarización y calidad al desarrollo de software. Con esto, se busca que las aplicaciones de software desarrolladas se adapten a las necesidades, expectativas y objetivos de las personas u organizaciones que las requieran (Christel y Kang, 1992). La ingeniería de software busca, en sus fases iniciales, un reconocimiento en profundidad 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 (Zapata y Arango, 2004).

Los métodos de desarrollo de software, tales como CDM (Oracle, 2000) y (Anderson y Wendelken, 1996), RUP (Kruchten, 1999), FDD (Coad y LeFebvre, 1999) y XP (Beck, 2000), entre otros, no utilizan criterios definidos para representar los problemas ni los objetivos y mucho menos la relación de consistencia entre ellos, pues toman como punto de partida la solución, que el interesado debe conceptualizar con el analista. Además, utilizan los objetivos y los problemas únicamente de manera descriptiva y

referencial, tratando de suministrar un contexto al desarrollo de software, pero sin incorporar esos elementos adecuadamente en la solución. El método de desarrollo de software UN-Método (Zapata *et al.*, 2006) realiza una aproximación a la relación entre problemas y objetivos utilizando el diagrama causa-efecto (Ishikawa, 1986) y el diagrama de objetivos de KAOS (Dardenne *et al.*, 1993) y Lamsweerde (2000). Pese a incluir diagramas para el análisis de problemas y objetivos, el analista debe construirlos de manera subjetiva, garantizando la consistencia entre sus elementos, lo cual es susceptible de incompatibilidades, al involucrar la posibilidad del error en un proceso manual.

El artículo plantea un método para establecer la consistencia de los problemas detectados en los procesos de educación de requisitos de software y su relación con los objetivos de la organización por medio de un conjunto de estructuras de problemas y objetivos que facilitan al analista su proceso de captura de requisitos en el desarrollo de aplicaciones de software.

El artículo se estructura así: en la sección 2 se presentan los antecedentes que dan cuenta de los dominios en los que se representan los objetivos y problemas; en la sección 3, se describe la metodología; en la sección 4 se representan las estructuras gramaticales de objetivos y problemas, así como las reglas de consistencia para su relación; en el capítulo 5 se muestra el prototipo de aplicación de las estructuras y, finalmente, las conclusiones y el trabajo futuro.

## 1. Antecedentes

### 1.1. Objetivos

Los objetivos constituyen los referentes principales para la toma de decisiones en una organización, ya que son sus ejes principales. Por tal motivo, debe existir coherencia en su representación y expresión, de manera que garantice tanto a interesados como a analistas su interpretación

y relación con otros componentes básicos de algún proceso de desarrollo de software que se inicie en la organización.

Navarro *et al.* (2000) plantean que los objetivos constituyen conocimientos que adquieren los estudiantes y que poseen una estructura conformada por un verbo, que puede representar un resultado, una acción o un comportamiento que denota un aprendizaje por parte del estudiante, un complemento con un tema que dé cuenta del área de conocimiento y unas circunstancias en las que se debe lograr. Los objetivos representan una estructura del estilo “el alumno debe ser capaz de”. Bloom *et al.* (1956) proponen un conjunto de verbos en tres áreas del conocimiento (cognoscitiva, afectiva y psicomotora) que ayudan a la redacción de dichos objetivos. Dicha taxonomía plantea problemas en el hecho de que no realiza diferenciación entre los verbos que expresan objetivos y aquellos que representan simplemente actividades.

Antón (1996; 1997; 1998) y Zapata y Lezcano (2009) presentan un conjunto de verbos que se pueden utilizar para la redacción de objetivos, clasificándolos en tres categorías (mantenimiento, mejoramiento y realización). Además, Zapata y Lezcano (2009) amplían los verbos mediante un análisis de las similitudes que presentan esos verbos. Todos estos trabajos hacen referencia exclusiva a los verbos, pero no se ocupan del resto de la frase que representa la totalidad de la estructura gramatical que constituye un objetivo.

Los diagramas para representar objetivos en diferentes dominios van desde el desarrollo de software (muy especialmente en la etapa de educación de requisitos), pasando por el análisis organizacional (donde se plantean desde diferentes puntos de vista) y se enfocan en diferentes contextos. Yu (1995) define  $I^*$  como un lenguaje orientado a objetivos que propone el uso de modelos dependiendo del nivel de abstracción de los objetivos planteados. Dardenne *et al.* (1993) presentan el diagrama de objetivos de KAOS como una manera de ligar gráficamente, y de forma jerárquica, los objetivos de alto nivel de una organización con los requisitos y expectativas que se generan alrededor del software.

Ambos diagramas se ocupan de la representación gráfica de los objetivos y otros elementos del dominio de un interesado, pero no presentan una estructura gramatical que permita deducir si lo que se incluye en la representación gráfica de un objetivo efectivamente lo es. Otra forma gráfica de representación de objetivos se realiza a través de esquemas preconceptuales, que permite la elaboración semiautomática del diagrama de objetivos de KAOS (Zapata *et al.*, 2007).

En el aspecto del análisis organizacional, Practical (1971; 1979), Sánchez (2006) y Ortegón *et al.* (2005) plantean, en la metodología de marco lógico, que los objetivos se describen como situaciones futuras a las que se desea llegar una vez se resuelvan los problemas, representando dichos objetivos en un árbol de objetivos. Kepner y Tregoe (2005; 1956) se refieren a los objetivos como estructuras que deben describir los estados que se busca alcanzar en procesos de toma de decisiones, de forma precisa, situándolos en el tiempo y en un contexto determinado. Estos trabajos presentan algunas pautas para tener en cuenta en las estructuras para la representación de objetivos, pero no se establecen estructuras gramaticales precisas que permitan establecer si una frase contiene o no un objetivo. Adicionalmente, en los ejemplos suministrados de estas metodologías en la literatura, la identificación de los objetivos es una responsabilidad que recae sobre los analistas.

## 1.2. Problemas

Cuando se presenta un grado de inseguridad para la organización y una brecha negativa entre lo planeado y lo ejecutado se dice que ocurre un problema, que puede conllevar a generar una posible solución que retorne la estabilidad requerida dentro de los diferentes procesos de la misma. Los problemas se clasifican en dos tipos, dependiendo de su naturaleza. En primer lugar, se cuentan los problemas que contienen toda la información precisa y se resuelven empleando un cierto algoritmo, dando y permitiendo una única solución correcta (problema cerrado) (López, 1989; Garrett, 1998). En segundo lu-

gar, aquellos que implican la existencia de una o varias respuestas en su solución, sin tener garantía total de plantear la solución correcta, ya que esta depende directamente de la interpretación de terceras personas (problema abierto) (López, 1989; Garrett, 1998).

Cuando se hace referencia a la representación de problemas, existen muchos dominios que la plantean, pero todos de acuerdo con su naturaleza y los tipos de problemas que se quiere representar. Carpenter (1985), en el ámbito matemático, demostró que para definir un problema existen diferentes variables, entre las que se destacan las de naturaleza lingüística, es decir, sintácticas, semánticas o una combinación de ambas. Entre estas variables se encuentra la cantidad de palabras, la secuencia de la información y la presencia de alguna palabra clave que pueda inferir la realización de alguna operación. Sin embargo, se considera que las variables semánticas son las más importantes. También en el campo matemático, Castro (1991) establece que el término “problema” involucra una proposición o enunciado, unos datos conocidos que hay que estudiar, una acción (que alguien debe averiguar), una meta u objetivo (por ejemplo, obtener un resultado) y un proceso (el modo de actuación para alcanzar el resultado).

En la ingeniería de software los diferentes métodos de desarrollo, tales como CDM Oracle (2000) y Anderson y Wendelken (1996), RUP, FDD y XP, no especifican estructuras que permitan la representación de problemas en los procesos de educación de requisitos, lo que implica que todo recae en la interpretación que el analista realice y en la forma como él represente y estructure los problemas.

Ortegón *et al.* (2005) plantean para el análisis organizacional, en la metodología de marco lógico que los problemas se representan como estados negativos y no como situaciones inexistentes. Se sugiere que la forma correcta de expresar los problemas se estructura como en el siguiente ejemplo: “las plagas destruyen la cosecha”, teniendo muy en cuenta que dicho problema no hace referencia a la inexistencia de algún elemento que dé cuenta del problema. Sugiere como

problema mal redactado el siguiente ejemplo: “No hay pesticidas”.

En otra metodología de análisis organizacional, Kepner y Tregoe (1965) representan los problemas como una brecha no cumplida entre un objetivo y una situación que ocurre actualmente en la organización y que está afectando algún proceso dentro de un área determinada. Se plantea que la estructura de los problemas se representa teniendo en cuenta cuatro dimensiones: identidad, ubicación, tiempo y magnitud, y se ligan con las preguntas qué, dónde, cuándo y cuánto.

Por otro lado, se plantea que un enunciado posee diferentes componentes que dan cuenta de la presencia dentro del mismo de una situación, un problema y una posible solución (estructura situación-problema-observación-solución-evaluación (Winter, 1997)).

En lo referente a los diagramas para la representación de problemas, se plantean dos estructuras básicas: Ishikawa (1986) plantea el diagrama causa-efecto, que permite detectar un problema central y las causas que lo ocasionan y Practical Concepts Inc. (1971; 1979), en la metodología de marco lógico, plantean un árbol de problemas. Ambos gráficos representan problemas y guías para su construcción, pero no estructuras gramaticales establecidas que permitan al analista enunciar el problema de forma automática, sino que aún requieren su experiencia y capacidad.

Al igual que en el caso del planteamiento de objetivos, en los trabajos anotados no se presentan estructuras gramaticales que puedan representar los problemas. Además, la identificación de los problemas recae directamente en el analista, que es el encargado de recopilarlos.

### 1.3. Relaciones entre objetivos y problemas

En los procesos de ingeniería de software —y especialmente en lo referente a la documentación, artefactos y diagramas que utilizan los métodos de desarrollo de software mencionados anteriormente— no se especifican estructuras y tampoco métodos para representar problemas ni obje-

vos; se deja esta tarea a la interpretación y la experiencia del analista. El método de desarrollo de software propuesto por la Universidad Nacional de Colombia —UN-METODO (Zapata *et al.*, 2006)— incorpora un conjunto de diagramas y artefactos para ayudar, apoyar y realizar el proceso de educación de requisitos en el desarrollo de software, priorizando el problema antes de pensar en la manera de solucionarlo. En este método se utilizan diagramas para la representación de objetivos (diagrama de KAOS) y problemas (diagrama causa-efecto) y además una forma de relacionarlos mediante una tabla explicativa de los procesos de la organización, que incluye los objetivos que busca alcanzar un proceso y los problemas que se ligan con ese proceso. Esta tabla se rige por reglas de consistencia, pero requiere todavía una alta dosis de interpretación del analista para elaborarla, puesto que no se definen las características que deberían cumplir los objetivos ni los problemas.

En el campo del análisis organizacional, Kepner y Tregoe (1965) plantean una relación secuencial entre la definición, el análisis de los problemas, la toma de decisiones y el planteamiento de objetivos de dicha solución. Se señala, eso sí, que todo el proceso recae directamente en el analista, quien es el encargado de determinar cuáles son los problemas, enunciarlos y tomar una decisión que elimine la situación negativa encontrada en la organización. Se analizan los problemas según cuatro dimensiones: identidad (¿Qué es lo que se trata de explicar?), ubicación (¿dónde se observa?), tiempo (¿cuándo ocurre?) y magnitud (¿cómo es de grave?). Estos autores parten de la generación de problemas para explicar la toma de decisiones. Para que exista un problema no solo se debe dar la condición de desequilibrio, sino que es necesario que alguien piense que la desviación es lo suficientemente importante como para corregirla.

Practical (2007; 1971) y Ortegón *et al.* (2005) plantean en la metodología de marco lógico una posible relación entre problemas y objetivos a través del árbol de problemas y del árbol de objetivos. El primero expresa un encadenamiento tipo causa-efecto: las condiciones negativas percibidas por los involucrados en relación con el problema en cuestión, en el cual se ordenan

los problemas principales permitiendo al analista identificar el conjunto de problemas sobre el cual se concentrarán los objetivos del proyecto. Esta claridad de la cadena de problemas permite mejorar el diseño, efectuar un monitoreo de los “supuestos” del proyecto durante su ejecución y, una vez terminado el proyecto, facilita la tarea del evaluador, quien debe determinar si los problemas se resolvieron (o no) como resultado del proyecto. El segundo diagrama es el árbol de objetivos, en el cual los problemas identificados en el árbol de problemas se convierten, como soluciones, en objetivos del proyecto como parte de la etapa inicial de diseñar una respuesta. Los objetivos identificados como componentes o productos de un proyecto se convierten en los medios para encarar el problema de desarrollo identificado y proporcionar un instrumento para determinar su impacto.

Cabe anotar que todavía el proceso depende directamente de la interpretación del analista, para definir qué es y qué no es un problema. Además, la relación entre problemas y objetivos se da en el plano de la solución y en ningún momento se refiere a objetivos organizacionales.

En ninguno de los métodos y metodologías descritas existe una estructura gramatical clara para establecer qué es un problema y qué no lo es y, además, no se estructura una forma de representarlos. Toda la interpretación corre por cuenta del analista y es él quien determina cuándo hay un problema y cómo se enuncia. Lo mismo ocurre con los objetivos, tanto de la solución como organizacionales: la relación entre objetivos y problemas es una actividad propia del analista, quien determina qué problemas provienen de un objetivo determinado o qué objetivos solucionan un problema específico.

## 2. Metodología

### 2.1. Fase de exploración

En esta fase se realiza una revisión de literatura y un análisis de las estructuras para expresar pro-

blemas y objetivos que se plantean en diferentes dominios, tanto en procesos de desarrollo de software como en procesos de análisis organizacional. Además, las formas de representación y la relación que pueden existir entre estos a fin de identificar los avances y las falencias que existen en esta temática y, muy especialmente, en los procesos de desarrollo de software (haciendo énfasis en la educación de requisitos).

Igualmente, se realiza una exploración de los métodos de ingeniería de software para determinar cómo estos llevan a cabo el proceso de captura de requisitos, procurando definir cuál es la tarea del analista y como él establece los problemas y los objetivos de la organización y su relación para determinar cuáles problemas deben resolverse de manera prioritaria mediante una aplicación de software.

También se hace una exploración de los diagramas para representar objetivos (diagrama de objetivos de KAOS y árbol de objetivos de marco lógico) y problemas (diagrama causa-efecto y árbol de problemas de marco lógico) a fin de determinar una posible relación de consistencia entre ellos que facilite y automatice el papel del analista dentro de la fase de educación de requisitos.

## 2.2. Fase de definición

Teniendo como base las estructuras de objetivos

y problemas planteadas en los diagramas de objetivos de KAOS y en el diagrama causa-efecto, y la fase de exploración realizada, se establece un conjunto de estructuras gramaticales que permiten expresar los objetivos y los problemas de forma coherente y siguiendo estructuras previamente definidas, así como un conjunto de reglas que faciliten la relación automática entre objetivos y problemas en los procesos de educación de requisitos.

## 2.3. Fase de elaboración

En esta fase se desarrolla un prototipo cuyas funcionalidades principales utilizan las reglas de consistencia entre objetivos y problemas.

## 3. Método

Zapata y Vargas (2011) plantean algunas estructuras gramaticales para representar objetivos, problemas y la relación entre ellos, que es aplicables al proceso de educación de requisitos de software, con el objetivo de mejorar la consistencia. En la tabla 1 se especifican algunas de las estructuras gramaticales definidas para enunciar los problemas. Las abreviaturas empleadas en dicha tabla son las siguientes: O=Oración, V=Verbo, Ad.=Adjetivo, SN=Sintagma Nominal, Adv.=Adverbio, S=Sustantivo.

Tabla 1. Estructuras gramaticales para enunciar problemas

| Caso | Descripción | Restricciones  | Ejemplos  |
|------|-------------|--|---|
| 1    | O→V+Ad+SN   | V→{en forma reflexiva impersonal o voz pasiva refleja, por ejemplo: "hay", "existe", "presenta"}<br><br>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.<br><br>Existen malas condiciones habitacionales en el conjunto residencial.  |
| 2    | O→V+SN+Adv  | V → {en forma reflexiva impersonal o voz pasiva refleja}<br><br>Adv. → {debe tener una connotación negativa, se sugieren palabras como: "a destiempo", "tardíamente", "equivocadamente", etc.}   | Se entregan las muestras a destiempo.   |
| 3    | O→V+Ad+SN   | V → {en forma reflexiva impersonal o voz pasiva refleja; se sugieren algunos como: "presentar", "haber", "cometer", "sobrepasar", "realizar", "generar", "utilizar"}<br><br>Ad. →{debe tener una connotación positiva; se sugieren palabras como "alto", "mucho", "demasiado", etc.}<br><br>SN → {debe tener una connotación negativa, por ejemplo: "demoras", "retrasos", etc.} | Se presenta mucha deserción escolar en la educación superior.<br><br>Hay demasiada accidentalidad de automotores en las vías.<br><br>Hay mucha basura acumulada en la calles<br><br>Se genera una alta tasa de accidentes en la intersección. |

En la tabla 2 se especifican algunas de las estructuras gramaticales definidas para enunciar los objetivos. Las abreviaturas empleadas en dicha tabla, son las siguientes: O=Oración, V1=Verbo, V2=Verbo, Ad.=Adjetivo, SN=Sintagma Nominal, Adv.=Adverbio, S=Sustantivo, C=Conjunción.

Tabla 2. Estructuras gramaticales para enunciar objetivos

| Caso | Descripción       | Restricciones   | Ejemplos  |
|------|-------------------|---|---|
| 1    | O→V1+Ad.+SN       | V1→{Verbo de logro}<br><br>Ad.→ {Debe tener connotación positiva. Por ejemplo: "Alta", "buena", "Adecuada"}   | Lograr alta producción de materia prima en la fábrica de telares.<br><br>Lograr buenas condiciones habitacionales en el conjunto residencial. |
| 2    | O→V1+C+V2+SN+Adv. | V1→{Verbo de logro}<br><br>C→{"que"}<br><br>V2 → {en forma reflexiva impersonal o voz pasiva refleja; se sugieren algunos verbos como: "presentar", "cometer", "realizar", "generar", "utilizar"}<br><br>Adv.→ {debe tener una connotación positiva; así, se sugieren palabras como: "a tiempo", "correctamente", etc.} | Lograr que se entreguen las muestras a tiempo.<br><br>Lograr que se realicen las actualizaciones correctamente.                               |

En la tabla 3 se presentan algunas de las reglas definidas para establecer las relaciones lingüísticas entre los objetivos y los problemas. Las abreviaturas empleadas en dicha tabla son las siguientes: O=Oración que expresa un objetivo, P=Oración que expresa un problema, V1=Verbo, V2=Verbo, Ad=Adjetivo, SN1=Sintagma Nominal, SN2=Sintagma Nominal, Adv1=Adverbio, Adv2=Adverbio, S=Sustantivo, C=Conjunción.

Tabla 3. Reglas definidas para relación entre problemas y objetivos

| Caso | Descripción                          | Restricciones   | Ejemplos   |
|------|--------------------------------------|---|--|
| 1    | O→V+Ad1+SN<br><br>P→V1+Ad2+SN        | V→{Verbo de logro}<br><br>V1 → {en forma reflexiva impersonal o voz pasiva refleja}<br><br>SN→ {Se usa el mismo sintagma nominal tanto para el objetivo como para el problema.}<br><br>Ad1 y Ad2→ {Los adjetivos deben poseer connotaciones contrarias.}    | Objetivo: Lograr alta producción de materia prima en la fábrica de telares.<br><br>Problema: Existe baja producción de materia prima en la fábrica de telares.<br><br>Objetivo: Lograr buenas condiciones habitacionales en el conjunto residencial.<br><br>Problema: Existen malas condiciones habitacionales en el conjunto residencial. |
| 2    | O→V+C+V1+SN+Adv1<br><br>P→V1+SN+Adv2 | V1 → {Se usa el mismo verbo conjugado tanto para el objetivo como para el problema.}<br><br>SN→ {Se usa el mismo sintagma nominal tanto para el objetivo como para el problema.}<br><br>Adv1 y Adv2→ {Los adverbios deben poseer connotaciones contrarias.} | Objetivo: Lograr que se entreguen las muestras a tiempo.<br><br>Problema: Se entregan las muestras a destiempo.  |

## 4. Prototipo

### 4.1. Descripción del prototipo

Aplicación de software desarrollada en Java que aplica las diferentes estructuras descritas para representar objetivos y problemas, y así permitir la automatización del proceso de educación de requisitos en lo que tiene que ver con la función del analista en la recopilación y enunciación de

los problemas y los objetivos y su representación en los diagramas.

En la figura 1 se observa la pantalla principal, la cual permite seleccionar entre estructuras de problemas y objetivos basados en las reglas definidas dentro del método establecido. La pantalla presenta dos opciones principales donde permite seleccionar entre objetivos y problemas dando la posibilidad de comenzar la construcción del proceso, ya sea con los problemas o con los objetivos.

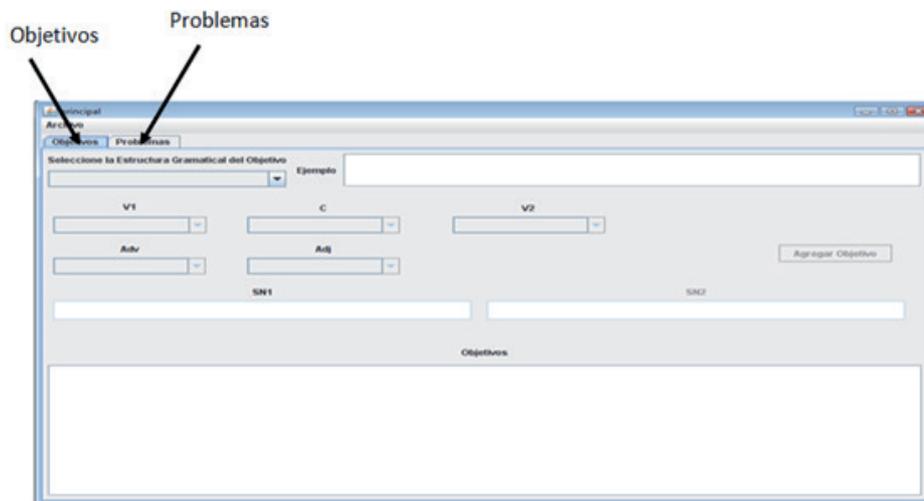


Figure 1. Pantalla principal del prototipo

## 5. Conclusiones

Diferentes dominios, como la ingeniería de software y la gerencia organizacional, requieren la identificación de problemas y objetivos y el establecimiento de una relación entre ellos para muchos de sus procesos, a fin de determinar qué problemas afectan el logro de un objetivo de la organización y, por ende, ayudar al buen desarrollo de la misma. El adecuado planteamiento de los objetivos y los problemas, así como las relaciones de consistencia entre ellos, pueden conducir a un mejor análisis de las soluciones problemáticas y, consecuentemente, al plantea-

miento de soluciones adecuadas, alineadas con la estrategia organizacional (sus objetivos) resolviendo las situaciones negativas (sus problemas).

En la literatura revisada no se especificó ninguna estructura gramatical que representase problemas ni objetivos, salvo el trabajo de Lezcano (2007), el cual plantea una estructura supremamente controlada y restringida basada en esquemas preconceptuales, pero no incluye algunas palabras importantes para calificar objetivos, tales como los adjetivos y los adverbios.

Tampoco se encontraron métodos automáticos que permitiesen relacionar objetivos y proble-

mas pues, pese a que algunos utilizan técnicas de representación tanto de objetivos como de problemas (diagrama de objetivos de KAOS, diagrama de objetivos de I\*, diagrama causa-efecto, árbol de problemas y árbol de objetivos de la metodología de marco lógico), sigue siendo un trabajo que depende del analista, basado en su intuición, sin que medie ningún chequeo de consistencia. Sin embargo, en la literatura revisada apareció un elemento común que pudo permitir el planteamiento de trabajo futuro en el área: se reconoció que los problemas son situaciones negativas y que su contraparte positiva se puede considerar un planteamiento de objetivos.

## Referencias

- [1] Anderson, C. y Wendelken, D. (1996). *The Oracle® Designer/2000 Handbook*. Addison-Wesley, Nueva York.
- [2] Antón, A. (1998) The use of goals to surface requirements for evolving system. *Proceedings of the 1998 (20th) International Conference on Software Engineering*, Kioto (Japón), 157-166.
- [3] Antón, A. (1997). *Goal Identification and Refinement in the Specification of Software-Based Information Systems*. Ph.D. Dissertation. Atlanta, Georgia Institute of Technology.
- [4] Antón, A. (1996). *Goal-Based Requirements Analysis*. *Proceedings of the Second IEEE International Conference on Requirements Engineering*, Colorado Springs (Estados Unidos), 136-144.
- [5] Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Boston, Addison-Wesley.
- [6] Bloom, B., Bertram, B., and Krathwohl, D. (1956). *Taxonomy of educational objectives: the classification of educational goals*, Handbook I: the cognitive domain. Nueva York, McKay Press.
- [7] Carpenter, T. P. (1985). *Learning to add and subtract: An exercise in problem solving*. En: E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives*, Erlbaum, Hillsdale, 17-40.
- [8] Castro, E. (1991). Resolución de problemas aritméticos de comparación multiplicativa. *Revista de la Sociedad Andaluza de Educación Matemática "Thales"*, (20):105-106.
- [9] Christel, M. y Kang, K. (1992). *Issues in requirements elicitation*. Technical Report CMU/SEI-92-TR-012. Software Engineering Institute, Carnegie Mellon University, Pittsburgh.
- [10] Coad, P. y LeFebvre, E. (1999). *Java Modeling in Color with UML*. Nueva York, Prentice Hall.
- [11] Dardenne, A., van Lamsweerde, A., y Fickas, S. (1993). *Goal-Directed Requirements Acquisition*. *Science of Computer Programming*, 20:3-50.
- [12] Garrett R.M. (1998). Resolución de problemas y creatividad: implicaciones para el currículo de ciencias. *Revista de investigación y experiencias didácticas*, 6(3), 224-230.
- [13] Ishikawa, K. (1986). *A guide to quality control*. Asian Productivity Organization, Tokio.
- [14] Kepner, Ch. y Tregoe, B. (1965). *The rational manager: a systematic approach to problem solving and decision making*. Nueva York, McGraw-Hill.
- [15] Kepner, Ch. y Tregoe, B. (1997). *The new rational manager: an updated edition for a new world*. Princeton, Princeton Research Press.
- [16] Kruchten, Ph. (1999). *Rational Unified Process—An Introduction*. Boston: Addison-Wesley.
- [17] Lamsweerde A. (2000). *Requirements engineering in the year 2000: a research perspective*. *Proceedings of the 22<sup>nd</sup> International Conference on Software Engineering*, Limerick (Irlanda), 5-19.
- [18] Lezcano L. (2007). *Elaboración semiautomática del diagrama de objetivos*. Tesis de Maestría, Universidad Nacional de Colombia, Medellín.
- [19] López, F. (1989). Dependencia-independencia de ampo y educación científica. *Revista de Educación*, (289):235-258.
- [20] Navarro, J., Valero-García, M., Sánchez, F., y Tubella, J. (2000). *Formulación de los objetivos de una asignatura en tres niveles jerárquicos*. *Memorias de las VI Jornadas sobre la Enseñanza Universitaria de la Informática JENUI 2000*, Alcalá (España), 457-462.
- [21] Oracle® Corporation (2000). *Oracle MethodSM CDM Quick Tour*. Oracle Corporation, Redwood City.
- [22] Ortegón, E., Pacheco, J. F., y Prieto, A. (2005). *Metodología del marco lógico para la planifica-*

- ción, el seguimiento y la evaluación de proyectos y programas. Santiago de Chile, Publicación de las Naciones Unidas.
- [23] Practical Concepts, Inc. (1979). The logical framework: a manager's guide to a scientific approach to design evaluation. Nueva York, Practical Concepts Inc.
- [24] Practical Concepts, Inc. (1971). The Logical Framework —Approach and training materials developed for the U.S. Agency for International Development. Washington, DC. Manuscrito sin publicar.
- [25] Sánchez, N. (2006). El marco lógico. Metodología para la planificación, seguimiento, y evaluación de proyectos. *Revista Visión Gerencial* 2(6):328-343.
- [26] Vargas, Fabio (2010). Método para establecer la consistencia de los problemas en el diagrama causa-efecto con el diagrama de objetivos de Kaos. Tesis para optar al título de Magíster en Ingeniería de Sistemas, Universidad Nacional de Colombia, sede Medellín.
- [27] Winter, E. O. (1977). A clause-relational approach to English texts: a study of some predictive lexical items in written discourse. *Instructional Science*, 6(1):1-92.
- [28] Yu, E. (1995). Modeling Strategic Relationships for Process Re-engineering. Tesis de Ph.D., Toronto, University of Toronto.
- [29] Zapata, C. M. y Lezcano, L. (2009). Caracterización de verbos usados en el diagrama de objetivos. *Dyna*, en prensa.
- [30] Zapata, C., y Vargas, F. (2009). Una revisión de la literatura en consistencia entre problemas y objetivos en ingeniería de software y gerencia organizacional. *Revista EIA* 11, 117-129.
- [31] Zapata, C.M.; Vargas, E.A. (2011). Innovación en el diseño y evaluación de proyectos: establecimiento de las relaciones lingüísticas entre objetivos y problemas. *Lámpsakos*, ISSN: No. 6, 46-55.
- [32] Zapata, C. M., Lezcano, A. y Tamayo, P. (2007). Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales. *Revista Escuela de Ingeniería de Antioquia*, 8:21-35.
- [33] Zapata, C., Villegas, S., y Arango F. (2006) Reglas de consistencia entre modelos de requisitos de UN-Método. *Revista Universidad Eafit*, 42(141):40-59.
- [34] Zapata, C., Gelbukh, E., Arango F. (2006). UN-Lencep: Obtención automática de diagramas UML a partir de un lenguaje controlado. *Avances en la Ciencia de la Computación*. 254-259.
- [35] Zapata, C. M. y Arango, F. (2004). Alineación entre metas organizacionales y elicitación de requisitos del software. *Dyna*, 71(143):101-110.