

DE REQUISITOS TEMPRANOS A REQUISITOS TARDÍOS: UN ENFOQUE BASADO EN METAS



AUTOR

Alicia Martínez Rebollar

Doctor (c) en Informática
Universidad de Trento, Italia
Docente en el Instituto Tecnológico de
Zacatepec, México e Investigadora invitada en
La Universidad Politécnica de Valencia, España
alimartin@dsic.upv.es
ESPAÑA-MÉXICO

AUTOR

Oscar Pastor López

Doctor en Informática
Director y Docente en el Departamento de Sistemas
Informáticos y Computación.
Universidad Politécnica de Valencia, España
opastor@dsic.upv.es
ESPAÑA

Fecha de recepción del artículo: 04 de Noviembre de 2005
Artículo Tipo 2

Fecha de Aceptación del Artículo: 18 de Noviembre de 2005

RESUMEN.

En la actualidad, la complejidad de los sistemas de información ha forzado a los ingenieros de software a contar con un profundo conocimiento del ambiente organizacional antes de iniciar con el desarrollo de un sistema de software. Por esta razón, en los últimos años la comunidad de ingeniería de software ha puesto una gran atención en la fase de requisitos tempranos. Actualmente, varios grupos de investigación trabajan en proponer técnicas para entender y representar el contexto organizacional en el cual trabajará el sistema de software. Sin embargo, sólo pocos estudios de investigación ofrecen un enfoque sistemático para llevar a cabo la equivalencia entre los modelos organizacionales y de requisitos. En este artículo se presenta un enfoque metodológico para derivar la funcionalidad del software a partir de modelos organizacionales. En el método propuesto, las metas organizacionales son la base para determinar las tareas relevantes que necesitan ser automatizadas. Se propone además un lenguaje de patrones para crear un nuevo modelo organizacional donde las tareas relevantes a ser automatizadas son delegadas hacia el actor sistema de software. El modelo organizacional generado a partir de este proceso representa al actor sistema de software dentro de su contexto organizacional. Esta propuesta permite dar un paso adelante en el proceso de incluir el modelado organizacional como una pieza clave en el proceso de producción de software.

PALABRAS CLAVES

Análisis de Metas,
Factores de Calidad,
Lenguaje de Patrones,
Modelos Organizacionales,
Requisitos Tempranos,
Requisitos Tardíos.

ABSTRACT

At present, the complexity of the information systems has forced the software engineers to have a deep knowledge of

the organizational environment before starting the software system development. This is why considerable attention has been given in recent years to the early requirements phase in the software engineering community. Currently, several research groups work in proposing techniques to understand and represent the organizational context in which the information system will work. However, only a few research studies offer a systematic approach for carrying out the equivalence between the organizational and requirements models. In this paper, a methodological approach for deriving the software functionality from organizational models is presented. In the proposed method, the organizational goals are the basis for determine the relevant tasks needed to be

automated. A pattern language is proposed to create a new organizational model where the relevant tasks to be automated are delegated towards the software system. The organizational model generated from this process represents the software system actor inside its organizational context (late requirements). By doing this, we go a step further in the process of including organizational modeling as a key piece in the software production process.

KEYWORDS

Goal Analysis,
Quality Factors,
Pattern Languages,
Organizational Modeling,
Early Requirements,
Late Requirements.

INTRODUCCIÓN

En la actualidad, la economía global competitiva, la demanda de una alta calidad en el desarrollo del software, así como un bajo costo y una rápida entrega, ha forzado a los ingenieros de software a plantearse seriamente el entendimiento profundo de la organización antes de iniciar la construcción de un sistema de software. Es por esta razón que en los últimos años la fase de requisitos tempranos ha recibido una gran atención por parte de los desarrolladores de software.

La fase de requisitos tempranos se enfoca al entendimiento de un problema mediante el estudio del contexto organizacional. La fase temprana es relevante porque un error de modelado en esta etapa puede acarrear costosos errores durante la fase tardía en el desarrollo de un sistema. No es sorprendente que existan múltiples trabajos de investigación que definan conceptos, lenguajes y técnicas de análisis en la fase temprana de requisitos [9][26][2][4][5].

Existan algunos trabajos de investigación que han tomado como punto de partida la fase de requisitos tempranos como la fuente la obtención de los requisitos tardíos [1][5][13][6][17][18]. La principal características de estas técnicas es el análisis y entendimiento de los procesos de negocios antes de la construcción de un sistema de información. La hipótesis de estos trabajos es que un proceso de elicitación de requisitos de software que no tome en cuenta el ambiente organizacional difícilmente podrá dar solución a las siguientes cuestiones:

- ¿Cuál es el valor añadido que dará el sistema de software a la empresa?
- ¿Cuáles son las tareas del negocio que deben automatizarse?
- La automatización de las tareas, ¿Es la mejor manera de resolver el problema que se tiene o que se pretende mejorar?
- ¿Los usuarios se verán beneficiados con la automatización de las tareas del negocio?

Estas preguntas sólo podrán resolverse si se cuenta con un profundo entendimiento de la organización que permita determinar cómo será utilizado el sistema de software, por

quiénes y en qué circunstancias deberá ser utilizado. Este conocimiento permitirá que el sistema proporcione un soporte real a los procesos del negocio. Sin embargo a pesar de las ventajas que aporta el enfoque de desarrollo de software basado en requisitos tempranos, existe una carencia de enfoques metodológicos que aborden en forma sistemática el desarrollo de un sistema de información, tomando como punto de partida los modelos de negocios y generando como resultado los requisitos del sistema de información. Para lograr que los enfoques de desarrollo basados en modelado organizacional sean efectivos en la práctica, es necesario que existan métodos y reglas que guíen al analista en el proceso de determinación de la información relevante en cada una de las etapas de modelado.

En este artículo, se propone un enfoque metodológico que pretende dar solución a esta carencia. El proceso de ingeniería de requisitos orientado a metas que se propone incluye un conjunto de pasos para identificar las tareas que requieren automatizarse así como un lenguaje de patrones que permite unir las fases temprana y tardía de requisitos. Nuestra propuesta hace énfasis en la representación de las diferentes alternativas para satisfacer las metas del negocio, así como en el análisis del impacto que tendrá la automatización de los planes en los factores de calidad esperados por la empresa. Esta propuesta permite obtener los requisitos funcionales de un sistema a construir a partir del entendimiento de las metas que dan soporte a todas las operaciones que se ejecutan en el negocio. Por otro lado, el lenguaje de patrones propuesto permite guiar la delegación de los planes a automatizar hacia un nuevo modelo organizacional que incluye, en forma explícita el sistema de información como un actor del modelo organizacional.

Este artículo se encuentra organizado de la siguiente forma: la sección 1 explica el modelado de metas y la metodología propuesta. La sección 2 y 3 detallan el método orientado a metas y el lenguaje de patrones que se proponen en este artículo. La sección 4 expone los trabajos relacionados con esta investigación, y finalmente en la sección 5 se presentan las conclusiones y trabajos futuros.

1. MODELADO DE METAS

Desde hace varias décadas, las metas han sido reconocidas como una herramienta básica en la ingeniería de requisitos [15]. Más recientemente, las metas han sido utilizadas para modelar requisitos tempranos [5] y requisitos no funcionales [7], los cuales han sido utilizados en muchas áreas de las ciencias computacionales. La importancia de las metas se hace evidente en ingeniería de requisitos, ya que proporcionan las motivaciones y razonamientos que permiten justificar cada uno de los requisitos del sistema de información. Es por esta razón que las metas han sido utilizadas como una herramienta que dan soporte en la construcción de sistemas de software.

En la literatura es posible encontrar distintas definiciones del concepto de meta:

- Son objetivos a ser satisfechos por el sistema analizado. Entendiendo "sistema" como el software a desarrollar junto con su ambiente operacional [10].
- Son los objetivos que el sistema analizado debería lograr [15].
- Son objetivos a lograr, los cuales proporcionan un marco de trabajo para un sistema deseado [3].
- Son una propiedad deseada para un ambiente [25].

En la mayoría de trabajos orientados a metas se remarca la importancia de las metas en el desarrollo del software [2][9]. Algunas de estas ventajas son:

Las metas permiten hacer explícita la relación entre las operaciones del negocio y las metas de alto nivel generalmente planteadas por los administradores.

Las metas proveen un criterio preciso para determinar la pertinencia de los requisitos. Un requisito es pertinente con respecto al conjunto de metas si su especificación es usada para satisfacer por lo menos a una de las metas.

Es posible utilizar las metas para determinar el proceso de negocio necesario para satisfacer cada meta.

El refinamiento de metas provee un mecanismo natural para estructurar documentos de requisitos complejos e incrementar su legibilidad.

Es posible utilizar las metas como un criterio de completitud y suficiencia en la especificación de requisitos. La especificación es completa con respecto al conjunto de metas si todas las metas pueden ser satisfechas con los requisitos determinados.

Es posible utilizar las metas para identificar y resolver conflictos entre los diferentes puntos de vista sobre la forma de satisfacer una meta.

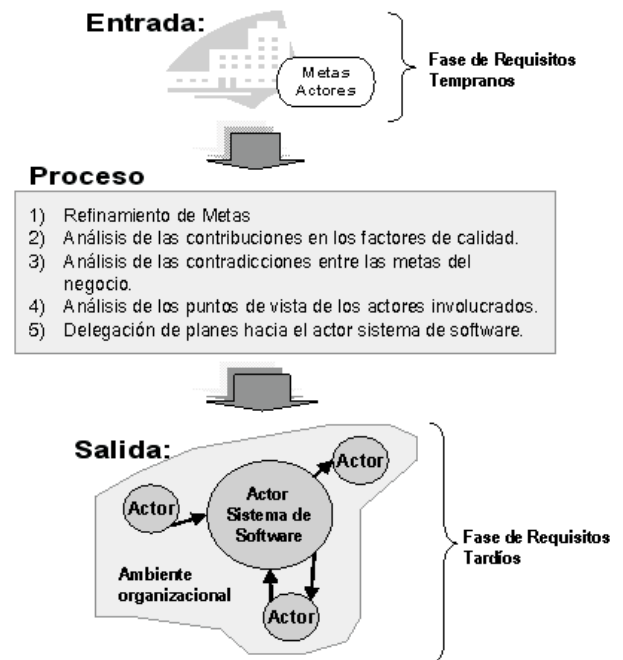
Sin embargo, a pesar todas las ventajas que proporcionan las metas, y de los múltiples trabajos llevados a cabo en esta área, aún existen factores que necesitan ser mejorados para asegurar su aplicación práctica. La siguiente subsección muestra nuestra propuesta de elicitación de requisitos de software orientado a metas que pretende dar respuesta a algunos problemas de los enfoques actuales de modelado de metas.

1.1 METODOLOGÍA PROPUESTA

La metodología de elicitación de requisitos propuesta en este artículo utiliza las metas organizacionales para descubrir los requisitos del sistema de software que se desea construir. Esta propuesta utiliza el Framework Tropos [5] para representar las metas organizacionales y los actores que intervienen en el negocio. Actualmente, Tropos ha sido utilizado para realizar análisis de metas [12], sin embargo, este framework no cuenta aún con un enfoque metodológico para guiar al analista en la construcción y el análisis de los modelos organizacionales con el objetivo de descubrir los planes que mejor satisfagan a las metas organizacionales. En este artículo se propone un enfoque metodológico para obtener requisitos funcionales a partir de modelos organizacionales descritos en Tropos, lo cual permite enfocar el proceso de obtención de los requisitos en el entendimiento de las metas que se persiguen en el contexto organizacional. Para realizar este proceso se

propone una técnica de análisis de metas y un lenguaje de patrones que permite unir la etapa temprana y tardía de requisitos de una forma sistemática.

Figura 1 Método de Elicitación de Requisitos orientado a metas



En la Figura 1 se presenta una vista general del método que se propone. La entrada de nuestra propuesta será el conjunto de metas de alto nivel de la organización, así como de los actores que intervienen en los procesos del negocio que se deseen automatizar con el sistema de información (fase de requisitos tempranos).

Una vez que las metas organizacionales han sido elicidadas y representadas en un modelo organizacional inicial, es necesario realizar el proceso propuesto en este artículo. Este proceso incluye un conjunto de análisis de metas que permiten identificar los planes que al ser automatizados, permiten satisfacer de la mejor manera las metas del negocio. En este proceso, se propone además un lenguaje de patrones que permiten guiar de una forma sistemática la construcción de un nuevo modelo organizacional que incluye al actor sistema de software. En este momento nos encontraremos en la fase tardía de requisitos, en el cual el sistema de software es descrito dentro de su ambiente operacional, junto con sus funciones y cualidades relevantes. Este nuevo modelo organización constituye el resultado final de la aplicación del método.

1.2 CASO DE ESTUDIO

Con el objeto de ilustrar el enfoque metodológico propuesto se ha utilizado un caso de estudio real de la empresa de desarrollo de software CARE Technology, S.A.

El caso de estudio analizado modela los procesos de negocio de la empresa RentaCar dedicada al alquiler de automóviles en la región de Alicante, España. Los procesos modelados fueron los relacionados con la compra y renta de automóviles, así como otras actividades derivadas tales como: mantenimiento y reparación de los automóviles, así como la renta de servicios extras (chofer, teléfonos, etc.). El desarrollo de este caso de estudio permitió analizar las diversas alternativas para seleccionar los planes que requieren automatizarse.

2. PROCESO DE ELICITACIÓN DE REQUISITOS BASADA EN METAS

Uno de los requisitos esenciales para resolver en forma efectiva el desarrollo de un sistema de software, es contar con herramientas que permitan representar el conjunto de metas que describen los estados deseados por la organización en la cual será utilizado el sistema. Las metas juegan un papel muy importante en la empresa, porque permiten dirigir todos los esfuerzos hacia un mismo objetivo. Sin embargo, el éxito de la satisfacción de las metas estará determinado por el grado de entendimiento que se tenga de éstas. Es necesario realizar un refinamiento de la metas para descubrir las distintas alternativas que existen para satisfacerlas. Este proceso permite tener un conocimiento más profundo de los planes en una organización y de las razones del por qué son ejecutadas dichos planes.

En esta propuesta se utiliza el Framework Tropos, el cual es considerado como una de las técnicas más conocidas y referenciadas en el área de modelado organizacional orientada al desarrollo de sistemas de software. Tropos ha adoptado los conceptos ofrecidos por el Framework i* [27], para soportar la etapa de modelado y análisis de requisitos tempranos; Tropos se define en términos de conceptos tales como: actores, dependencias sociales entre actores, dependencias de metas, metas suaves, planes y recurso. Este framework utiliza dos diagramas para representar los modelos organizacionales: el diagrama de actores y el diagrama de metas. En estos modelos las metas organizacionales juegan un papel fundamental. El diagrama de actores se utiliza para representar las dependencias que existen entre los actores para satisfacer sus metas; mientras que el modelo de metas se utiliza para representar en forma detallada, el comportamiento que cada actor realiza para satisfacer sus metas y lograr satisfacer las dependencias con otros actores.

Tropos clasifica las metas como Hardgoal y Softgoal, el primer tipo nos ayudará a identificar los aspectos funcionales del software que se desea construir. Por su parte, las Softgoals nos permitirán enfocarnos a los aspectos no funcionales del sistema de software a desarrollar.

De esta manera teniendo como soporte el Framework Tropos se propone un método bien definido que guíe el proceso de ingeniería de requisitos para la obtención de los requisitos de un sistema de software utilizando un enfoque orientado a metas que permita evaluar las alternativas que mejor satisfagan las metas del negocio. El método propuesto tiene como punto de partida el modelo de actores en el cual se

muestran gráficamente los actores que intervienen en el negocio, sus metas y sus dependencias. En la Figura 2 se muestra una vista parcial del diagrama de actores para el caso de estudio RentaCar. El diagrama de actores inicial es refinado en el modelo de metas. En las siguientes subsecciones se explica el procedimiento para refinar las metas, así como también se detallan los análisis llevados a cabo en las metas con el fin de identificar los planes que requieren automatizarse.

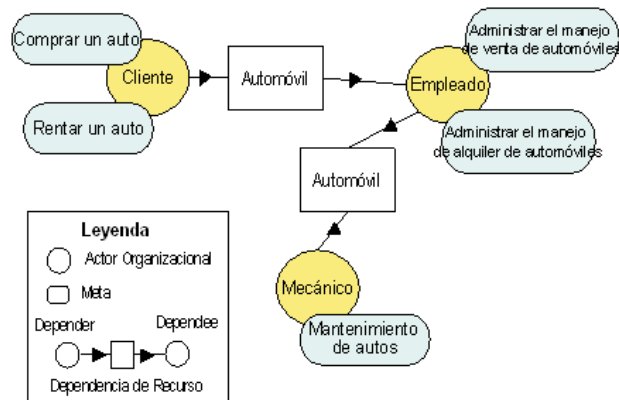
2.1 REFINAMIENTO DE METAS

Los mecanismos de refinamiento de metas han sido discutido en la literatura por varios autores [9] [2][24][7][17][25]. En Tropos [12] el refinamiento de metas es llevado a cabo a través de las ligas AND-OR Decomposition.

El primer paso de nuestra propuesta es el proceso de refinamiento de metas (Hardgoal) utilizando el diagrama de actores. Este refinamiento servirá para crear el diagrama de metas del Framework Tropos. El refinamiento se lleva a cabo de la siguiente manera: Cada actor identificado en el diagrama de actores puede tener de 1 a n metas generales.

Una meta general refleja los estados deseados por un actor en el cumplimiento de sus procesos del negocio. Estas metas se muestran gráficamente en el modelo de actores. Posteriormente cada una de estas metas son refinadas hasta obtener operaciones que las satisfagan. Este refinamiento detallado se realiza en el modelo de metas.

Figura 2 Vista parcial del diagrama de actores del caso de estudio: RentaCar

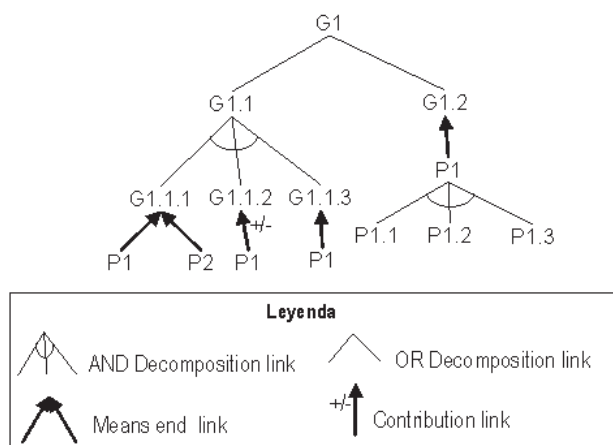


Cada meta general G1 se refinará utilizando las ligas: AND/OR decomposition, means-end o contribuciones. Las ligas AND decomposition se utiliza para representar el conjunto de submetas (G1.1, G1.2 ...) que harán posible satisfacer la meta G1. Las ligas OR decomposition se utilizan para representar las alternativas que pueden tenerse para satisfacer la meta G1. Las ligas means-end se utilizan para refinar una meta G1 en varias submetas (G1.1, G1.2,...) o varios subplanes (P1.1, P1.2,...). La liga de contribución generalmente se utiliza para especificar la contribución de una meta o un plan hacia un Softgoal.

El proceso de refinamiento se realiza en cada una de las metas y submetas obtenidas hasta que se alcanza un nivel en el que las metas pueden ser satisfechas a través de un conjunto de planes concretos. En este caso se utilizarán ligas means-end o ligas de contribución las cuales permiten refinar una meta en un conjunto de planes. Los planes a su vez pueden refinarse en otros subplanes utilizando las ligas AND/OR decomposition. En la Figura 3 se muestra un esquema de este refinamiento de metas y planes.

Durante el proceso de refinamiento de metas es posible delegar ciertas actividades a otro actor organizacional. También es posible detectar los casos donde se requiere que otro actor organizacional ejecute una acción o provea un recurso. Para representar estos casos, en Tropos se utiliza el concepto de dependencia.

Figura 3 Refinamiento de Metas

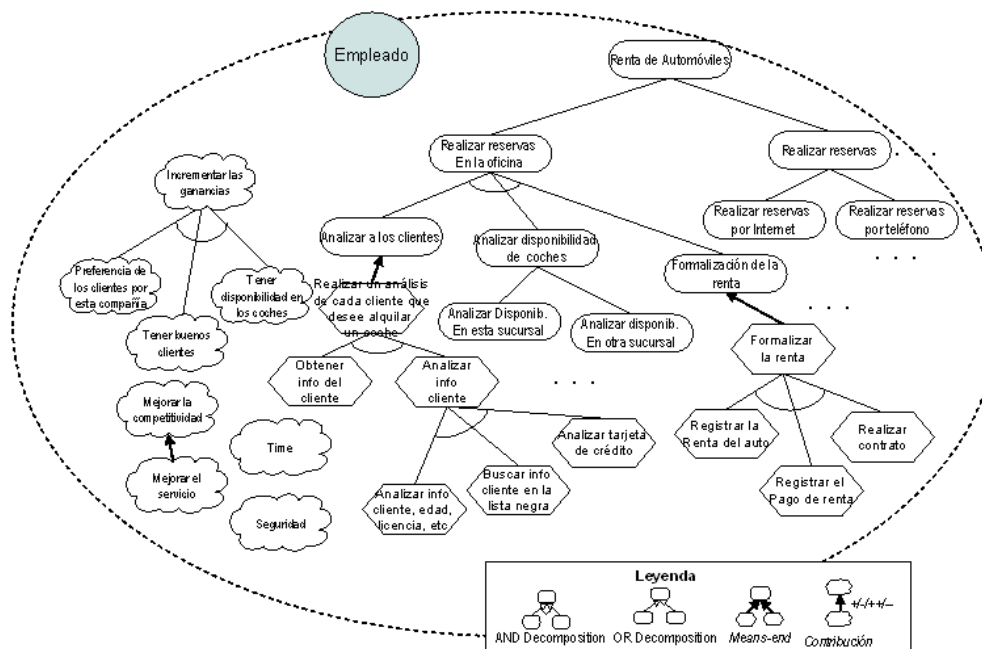


Las dependencias detectadas en el modelo de actores deben ser las mismas que en el modelo de metas. De esta forma se asegura que exista trazabilidad entre ambos modelos.

El refinamiento de metas suaves (Softgoals) se realiza de la misma forma que el refinamiento de metas Hardgoal, la única diferencia radica en que el caso de las metas suaves, no es posible determinar con precisión la satisfacción de las metas, ya que su refinamiento no se finaliza con la operacionalización de planes, sino con la determinación de planes que ayudan a satisfacer en cierto grado la meta que esta siendo refinada. En nuestro análisis, las metas suaves son consideradas como los factores de calidad que ayudarán en el incremento de mejora de los planes del negocio, así como en el incremento de la calidad en el producto final obtenido. Algunos de los factores de calidad que pueden ser considerados en el análisis de metas son: Seguridad, desempeño, usabilidad, flexibilidad, interoperabilidad, etc. [14].

Una vez que se ha realizado el refinamiento de metas, y que los factores de calidad deseados por la empresa han sido identificados, el siguiente paso estará enfocado en el análisis de contribuciones de los planes existentes con los factores de calidad seleccionados. Por cuestiones de espacio, en la Figura 4 sólo se muestra una vista parcial del refinamiento de metas del actor Empleado del caso de estudio RentaCar. En este ejemplo, una de las metas del actor es la renta de automóviles, esta meta se refina en dos submetas: (1) Realizar reservas directamente en la oficina y (2) Realizar reservas través de medios alternativos, como puede ser por Internet o por vía telefónica. La meta de realizar reservas directamente en la oficina es descompuesta en tres submetas unidas con una liga AND: Analizar los clientes, Analizar la disponibilidad de los coches y la Formalización de las metas. Los factores de calidad deseados por la compañía en este caso de estudio son: Competitividad, Tiempo y Seguridad.

Figura 4 Vista parcial del modelo de metas del caso de estudio RentaCa



2.2 ANÁLISIS DE LAS CONTRIBUCIONES EN LOS FACTORES DE CALIDAD

Una contribución describe la influencia que tiene la satisfacción de una meta en la satisfacción de otra meta o tarea del negocio. Las contribuciones pueden ser positivas "+", negativas "-", contribución suficiente "++", y parcialmente denegada "--" [12].

El segundo paso de nuestra propuesta es el análisis de contribuciones, el cual tiene como objetivo identificar los planes que mejor satisfacen a los factores de calidad. A continuación se muestran los pasos para llevar a cabo el análisis de contribuciones:

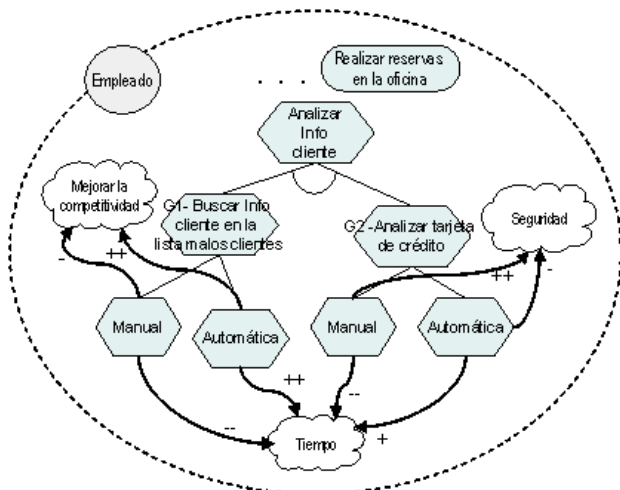
Paso 1. Propagación de los planes finales. En esta propuesta llamaremos planes finales a aquellos planes que se encuentren en las hojas de la estructura de refinamiento de metas. La propagación se realiza descomponiendo cada plan final en dos subplanes que representarán la manera de ejecutar los planes. Las dos alternativas posibles son Manual o Automática.

Paso 2. Contribución a los factores de calidad. Para cada una de los planes creados en el punto anterior, se coloca una liga de contribución hacia los factores de calidad (Softgoals) del modelo de metas.

Paso 3. Análisis de las contribuciones. Para llevar a cabo este análisis se requiere comparar las contribuciones de los planes que han sido propagados en el paso 1, y los factores de calidad del negocio. La comparación se realiza con los valores (++/--/-/+) colocados en las ligas de contribuciones. Este paso permitirá identificar los planes que mejor satisfacen los factores de calidad a través de las contribuciones.

Continuando con el caso de estudio, en la Figura 5 se muestra una vista parcial de la propagación de los planes del actor Empleado: Buscar información del cliente en la lista negra, y Analizar tarjeta de crédito del cliente. Cada uno de estos planes se propaga en dos subplanes: buscar la información del cliente de forma manual y buscar al cliente de forma automática.

Figura 5 Contribuciones a los factores de calidad



Posteriormente se crean ligas de contribución de estos planes hacia los factores de calidad. Finalmente se analizan estas contribuciones para seleccionar el plan que mejor satisfaga al factor de calidad, por ejemplo el plan buscar al cliente de forma automática posee una contribución suficiente (++) al factor de calidad: Tiempo (tomando en cuenta el factor de calidad como ahorro de tiempo). La ejecución manual de esta actividad tiene una contribución negativa en el factor de calidad Tiempo. Sin embargo, no en todos los casos resulta trivial deducir el plan que debe automatizarse, por ejemplo, en el caso del plan: Analizar tarjeta de crédito del cliente, las ligas de contribución resultan contradictorias con respecto a los factores de calidad: Tiempo y Seguridad. En estos casos se requieren análisis adicionales que serán explicados a continuación.

El análisis de las contribuciones además de identificar los planes que requieren ser automatizados, permite identificar las contradicciones al satisfacer las metas del negocio. La siguiente subsección explica el procedimiento para analizar estas contradicciones.

2.3 ANÁLISIS DE LAS CONTRADICCIONES ENTRE LAS METAS DEL NEGOCIO

El tercer paso de nuestra propuesta es el análisis de las contradicciones entre las metas del negocio.

La selección de los planes que requieren automatizarse para satisfacer las metas organizacionales no es una tarea trivial. En algunas ocasiones, los empleados de la organización no tienen una idea clara de la mejor forma de satisfacer las metas generales de la empresa. Esto es ocasionado por el desconocimiento del proceso global del que forman parte. Los análisis llevados a cabo anteriormente ayudan a contar con una visión global de la empresa y de las alternativas que pueden originarse para alcanzar las metas organizacionales a través de la implantación de un sistema de software. Otra ventaja del análisis de metas propuesto es que nos permitirá evaluar de manera global los objetivos de la empresa y de cómo ellos son alcanzados.

A continuación se describen los pasos necesarios para analizar contradicciones entre las metas del negocio:

Paso 1. Crear una matriz con los planes finales y los factores de calidad. Los planes colocados en esta matriz serán aquellos planes que se desean analizar para determinar si deben ser automatizados.

En la Tabla 1 se muestra un ejemplo de nuestro caso de estudio, en el cual se analizan los subplanes propagados en manual o automática. Los planes analizados son: Plan 1: Buscar información del cliente en la lista negra (ejecutadas de forma manual, o de forma automática) y Plan 2: Analizar la tarjeta de crédito (ejecutadas de forma manual, o de forma automática). En la parte superior de la tabla se colocan los factores de calidad que se desean alcanzar: Competitividad, Reducción de Tiempo y Seguridad.

Tabla 1 Matriz de contribuciones

Planes	Competitividad	Tiempo	Seguridad
Plan 1 (Manual)	-	--	
Plan 1 (Automática)	++	++	
Plan 2 (Manual)	-	--	++
Plan 2 (Automática)	+	++	-

Plan 1: Buscar información del cliente en la lista negra.

Plan 2: Analizar la tarjeta de crédito.

Paso 2. Colocar las contribuciones de los planes y factores de calidad en la matriz.

Paso 3. Comparar los subplanes propagados que corresponde al mismo plan. Continuando con nuestro caso de estudio (Tabla 1), existen dos formas alternativas para satisfacer el plan 1: manual o automática. Por este motivo, es necesario comparar las contribuciones de estos dos planes con el objetivo de identificar el plan que mejor satisfaga las metas del negocio. Por ejemplo, la ejecución automática del plan 1 reportaría más contribuciones positivas que la ejecución manual de este plan.

Paso 4. Identificar contradicciones entre las contribuciones de los factores de calidad. Al realizar la comparación de los subplanes es posible descubrir contradicciones en la satisfacción de los factores de calidad. Una opción para resolver las contradicciones es definir prioridades entre los factores de calidad. Otra opción es, analizar los actores involucrados en la ejecución de los planes (este proceso se analiza en la siguiente subsección).

En la Tabla 1 se muestra un ejemplo de las contradicciones que pueden presentarse durante la determinación de contribuciones. Al analizar las contribuciones del plan 2, pudiera parecer más conveniente ejecutar el plan de forma automática, sin embargo al comparar el plan con el factor de seguridad, la contribución resulta negativa. En este caso específico resulta más conveniente ejecutar la tarea en forma manual.

Para resolver los conflictos, es necesario que el analista determine la prioridad de los factores de calidad para elegir si la tarea requiere automatizarse.

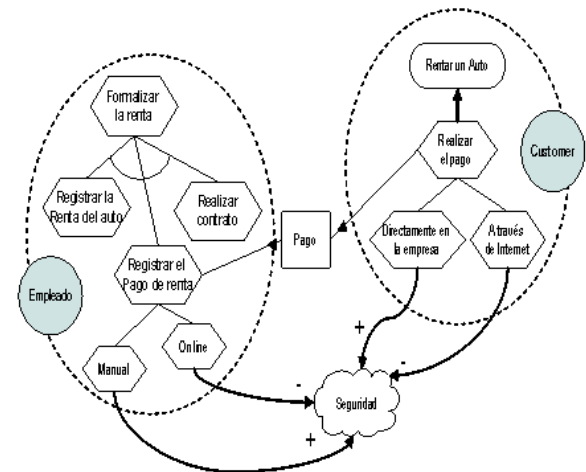
2.4 ANÁLISIS DE LOS PUNTOS DE VISTA DE LOS ACTORES INVOLUCRADOS

El cuarto paso de nuestra propuesta es el análisis de los puntos de vista de los actores involucrados en la satisfacción de una meta. La satisfacción de las metas de un actor puede verse afectada no sólo por la ejecución de sus propios planes, sino por la ejecución de los planes de otro actor.

El framework Tropos utiliza un enfoque orientado a dependencias para modelar la organización. Una dependencia es una liga entre dos actores que indica que un actor depende de otro para lograr sus objetivos. El actor dependiente se

denomina Dependier, y el actor de cual se depende se denomina Dependee. El objeto en el cual se enfoca la dependencia se denomina Dependium. En Tropos, el dependium puede ser un plan, una meta o un recurso [28].

Las relaciones de dependencia permiten que un actor alcance sus metas mediante la colaboración con otros actores organizacionales. En esta propuesta se analizan las dependencias entre los actores con el fin de facilitar la toma de decisiones de los planes que requieren automatizarse. Por lo tanto, la toma de decisiones estará basada no sólo en las contribuciones de los planes de cada actor, sino también en las contribuciones de los actores involucrados a través de una relación de dependencia.

Figura 6 Analizando los puntos de vista de los actores involucrados

La Figura 6 muestra un ejemplo del análisis de los puntos de vista de los actores involucrados con el objetivo de determinar la forma más conveniente para realizar el pago del alquiler de un automóvil (actor Empleado). Se analizan las contribuciones del actor cliente, el cual se encuentra involucrado en este plan a través de una dependencia de recurso (Pago). En este caso si el factor de Seguridad tiene una alta prioridad y si ninguno de los actores involucrados contribuye positivamente en este factor, la opción más apropiada puede ser ejecutar el plan en forma manual.

2.5 DELEGACIÓN DE PLANES HACIA EL ACTOR SISTEMA DE SOFTWARE

El último paso de nuestra propuesta es la delegación de los planes (que requieren automatizarse) hacia el actor sistema de software.

Una vez que los análisis de metas, contribuciones y dependencias han sido realizados, se tendrán identificados los planes que mejor satisfacen las metas organizacionales de alto nivel, y que mejor contribuyen con los factores de calidad deseados por la empresa. Estos planes constituyen los requisitos que deben ser considerados en la construcción del sistema de información. El objetivo de este paso es agrupar todo los planes a automatizar en un actor que represente al

sistema a desarrollar. Por lo tanto, se construirá un nuevo modelo organizacional que muestre todas las interacciones entre los actores organizacionales y el actor sistema software. La delegación de los planes se realiza de forma sistemática a través de un lenguaje de patrones, el cual será explicado en la siguiente sección.

3. LENGUAJE DE PATRONES

El lenguaje de patrones que se propone en este artículo es utilizado para realizar la delegación de los planes que requieren automatizarse hacia el actor sistema de software. Este lenguaje lo hemos denominado FELRE por sus siglas en inglés (From Early Requirements to Late Requirements).

Actualmente los lenguajes de patrones han sido ampliamente utilizados como herramientas para definir enfoques metodológicos que guíen el proceso de modelado y diseño [11][8][23]. En esta investigación se ha utilizado esta técnica con el objetivo de proporcionar un método bien definido y preciso que permita unir las fases temprana y tardía de requisitos de una forma sistemática. La utilización de este lenguaje permite obtener un nuevo modelo organizacional que incluye el sistema a construir como un actor organizacional, el cual permitirá visualizar el sistema junto con sus interacciones en su medio operacional. La inclusión del actor sistema de software dentro del ambiente operacional permite contar con una descripción de alto nivel de los planes que deben ser soportados por un sistema de software. Estos planes ayudan a la satisfacción de las metas organizacionales, así como al cumplimiento de los factores de calidad deseados.

De esta forma, el análisis del nuevo modelo organizacional está enfocado en los planes a ser automatizados. Este nuevo modelo cuenta con el nivel de abstracción necesario para iniciar el proceso de elicitación de los requisitos del sistema de información. En [20] se describe el proceso para la identificación de casos de uso a partir del modelo que incluye al actor sistema de software, y en [21] se presenta la propuesta para generar un esquema conceptual a partir de este modelo organizacional.

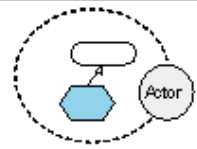
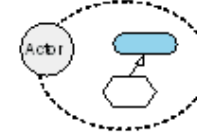
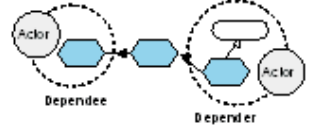
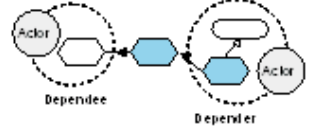

El lenguaje FELRE identifica las distintas posibilidades que pueden presentarse al desear automatizar un plan en el contexto organizacional. Estas posibilidades se encuentran explicadas a través de los cinco patrones de automatización de los que está compuesto el lenguaje. En la Tabla 2 se explican cada uno de los patrones, además de los casos en los que estos pueden ser utilizados. El algoritmo utilizado por el lenguaje de patrones FELRE para la delegación de los planes hacia el actor sistema de software se muestra a continuación:

Paso 1. Identificar los planes que se pretenden automatizar. Este paso se lleva a cabo con los análisis de las metas descritos en la sección anterior.

Paso 2. Colocar el actor sistema de software en el nuevo modelo organizacional, así como los actores, que tengan algún plan, meta o relación de dependencia que se pretenda automatizar.

Paso 3. Delegar los planes y metas que se pretenden automatizar hacia el actor sistema de software. Para este

Tabla 2 Breve descripción de los patrones del Lenguaje FELRE

Nombre del Patrón	Utilización	Plan o Meta a ser automatizada
1) Patrón de Automatización: <i>Plan final</i> sin dependencias	Este patrón será utilizado cuando un <i>Plan final</i> sin dependencias necesite ser automatizado. <i>Planes Finales</i> sin dependencias son aquellos planes finales que no requieren la intervención de otros actores para satisfacerse.	
2) Patrón de Automatización: <i>Plan General</i> o <i>Meta General</i>	Este patrón será utilizado cuando un <i>Plan General</i> o una <i>Meta General</i> necesiten ser automatizados.	
3) Patrón de Automatización: Planes <i>Depender-Dependee</i> Actor	Este patrón será utilizado cuando los planes a ser automatizados sean tanto del actor <i>Depender</i> como del actor <i>Dependee</i> .	
4) Patrón de Automatización: <i>Plan del Actor Dependee</i>	Este patrón será utilizado cuando el plan del actor <i>Depender</i> necesite ser automatizado.	
5) Patrón de Automatización: <i>Plan del Actor Dependee</i>	Este patrón será utilizado cuando el plan del actor <i>Dependee</i> necesite ser automatizado.	

paso, es necesario realizar lo siguiente:

Paso 3.1 Analizar los planes de cada actor organizacional. La especificación de un actor puede contener varias metas y planes, los cuales llegan a formar una estructura arborescente. Se realiza un recorrido infijo en cada uno de los actores organizacionales para identificar los planes que requieran ser automatizados. Este proceso continua hasta que todos los planes y metas de cada actor han sido recorridos.

Paso 3.2 Analizar cada uno de los elementos internos obtenidos en el recorrido infijo. Cuando se presente un elemento que tenga asociada una relación de dependencia, todos los elementos de esa relación deben ser analizados (dependee, dependee y dependum).

Este análisis permite identificar los patrones de automatización que se presentan en el modelo organizacional. Los patrones pueden ser:

- (1) Patrón de Automatización: Plan final sin dependencias
- (2) Patrón de Automatización: Plan General o Meta General
- (3) Patrón de Automatización: Planes Dependee-Dependee Actor
- (4) Patrón de Automatización: Plan del Actor Dependee
- (5) Patrón de Automatización: Plan del Actor Dependee.

Paso 3.3 Una vez identificado el patrón que se requiere utilizar, se siguen los pasos descritos en ese patrón para delegar los planes y metas que se desean automatizar al actor sistema de software.

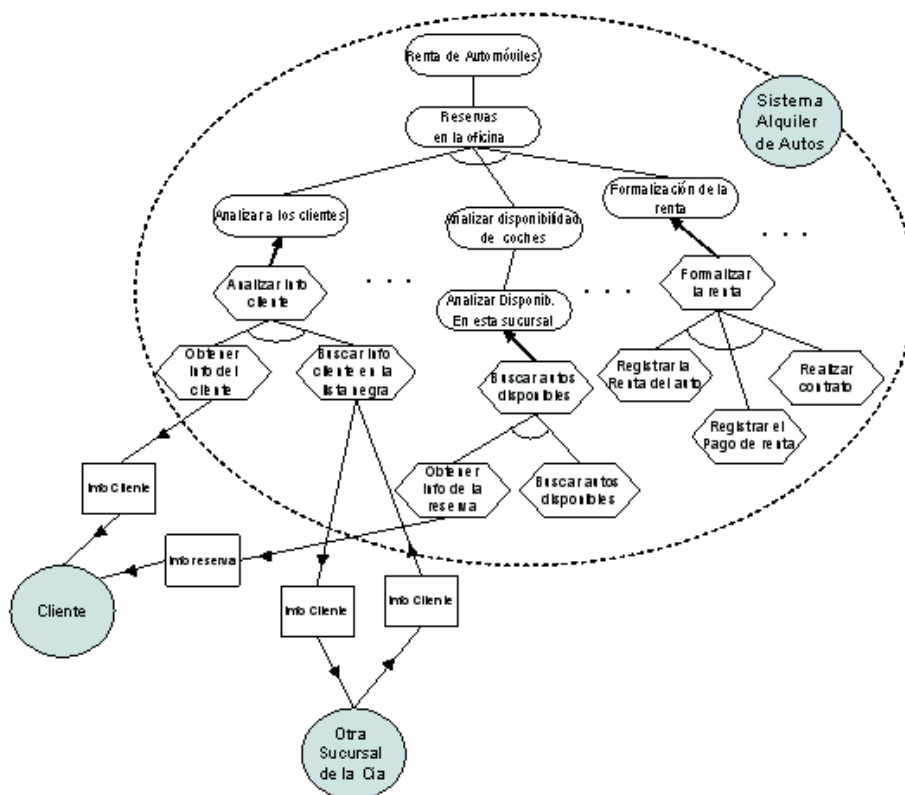
describe detalladamente cada uno de los patrones que componen el lenguaje. En [19] se muestra la descripción completa de estos patrones de automatización.

Una vez que se ejecuta el lenguaje de patrones se obtiene un modelo organizacional que incluye al sistema de software, además de todos los demás actores organizacionales que interactúan con el sistema. En la Figura 7 se presenta una vista parcial del modelo organizacional para el caso de estudio Rentacar. El modelo representa los requisitos del sistema que se pretende construir, así como las entradas y salidas necesarias por el sistema. De esta manera se puede garantizar que el sistema a construir satisficiera las metas organizacionales, además de que ayudará a alcanzar los requisitos de calidad esperados por el negocio.

4. TRABAJOS RELACIONADOS

Actualmente existen múltiples trabajos que utilizan las metas durante el proceso de generación del modelo de requisitos. Entre los trabajos más relevantes se encuentra el método orientado a metas: KAOS [17][16][9], el cual proporciona reglas formales para derivar los requisitos basados en teorías en lenguajes de especificación formal. Este método permite analizar tanto los requisitos funcionales como los no funcionales. Sin embargo, la utilización de este método está restringida a analistas familiarizados con notaciones y métodos formales. Otra desventaja de KAOS es que proporciona poco soporte a la selección de alternativas, actividad que es indispensable para tomar decisiones de diseño sobre las tareas que necesitan ser automatizadas.

Figura 7 Vista parcial del modelo organizacional con el actor sistema de software



Otro método orientado a metas es GBRAM (Goal Based Requirements Analysis Method) [24] [2], el cual está enfocado a la derivación de los requisitos operacionales a partir de estas metas. Sin embargo, en este método no existe una distinción entre la información de la fase de requisitos tempranos y la fase de requisitos tardíos[26]. Esto ocasiona que no cuente con una representación clara del proceso completo de producción de un sistema de software.

Otros trabajos se han enfocado en el estudio de las metas analizando el impacto en los requisitos no funcionales, tal es el caso del Framework NFR propuesto por Chung [7].

La diferencia de nuestro enfoque con los trabajos arriba expuestos radica en la propuesta de un método de análisis que guíe al analista en la construcción de un sistema de información, haciendo énfasis en las fases temprana y tardía de requisitos, y proporcionando además un lenguaje de patrones que ayude a la unión de estas dos fases del proceso de generación de sistema de software.

5. CONCLUSIONES Y TRABAJOS FUTUROS

En este artículo se describe un enfoque metodológico para identificar requisitos de software. Este método está formado por dos procesos fundamentales: análisis de metas y aplicación de patrones de automatización.

En el primer proceso tiene como objetivo identificar los planes que requieren automatizarse. Este proceso se basa en la determinación de las tareas que permiten satisfacer de la forma más conveniente las metas organizacionales. En este proceso orientado a metas se identifican además los factores de calidad que la empresa desea lograr con la implementación de un sistema, analizando las contradicciones y contribuciones de los planes con los factores de calidad deseados. Esta fase de análisis concierne con la etapa temprana de requisitos.

El segundo proceso de nuestra propuesta consiste en delegar los planes identificados para su implementación a un nuevo actor organizacional que representa al sistema de software. Este proceso se realiza a través de un lenguaje de patrones propuesto que permite guiar de forma sistemática la construcción de un nuevo modelo organizacional. El nuevo modelo organizacional permite visualizar las interacciones del actor sistema de software con los actores del contexto organizacional en el cual será utilizado (en este punto nos encontraremos en la fase tardía de requisitos). De esta manera, el lenguaje de patrones propuesto permite unir las fases temprana y tardía de requisitos.

El trabajo de investigación presentado en este artículo forma parte del proyecto PASMO (Producción Automática de Software a partir de Modelos Organizacionales). Actualmente se trabaja en la mejora de las reglas para utilizar el modelo organizacional para generar un modelo de requisitos, así como la generación del modelo conceptual especificado en OOMethod[22].

6. RECONOCIMIENTOS

Este trabajo de investigación ha sido parcialmente financiado por el proyecto MEC con referencia TIN2004-03534, en la Universidad Politécnica de Valencia, España y el programa SUPERA, en México.

7. REFERENCIAS

- [1] Alencar F., Pedroza F., Castro J., and Amorim R. New Mechanism for the Integration of Organizational Requirements and Object Oriented Modeling. Proceeding of the VI Workshop on Requirements Engineering (WER 2003). Piracicaba SP, Brasil, 2003. pp 109-123.
- [2] Antón I. Annie. Goal Identification and Refinement in the Specification of Software-Based Information Systems. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, June 1997.
- [3] Anton Annie. Goal based requirements analysis. In Proceeding of 2nd International Conference on Requirements Engineering ICRE'96, Colorado, USA, 1996. Pages 136-144.
- [4] Bider Llia, and Khomyakov Maxim. If you wish to change the world, Start with yourself. An Alternative Metaphor Objects Interaction. 4st International Conference on Enterprise Information Systems (ICEIS 2002), Ciudad Real, Spain 2002. Vol. 2, pp.732-742.
- [5] Bresciani P., Giorgini P., Giunchiglia F., Mylopoulos J., and Perini A., TROPOS: An Agent-Oriented Software Development Methodology. In Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers. May 2004; Volume 8, Issue 3, Pages 203 236.
- [6] Bubenko J.A., Jr, D. Brash and J.Stirma. EKD User Guide. Dept. of Computer and System Science, KTH and Stockholm University, Sweden ESPRIT Project No. 22927, ELEKTRA, Document, feb 5, 1998.
- [7] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers 2000.
- [8] Coad P., North D., and Mayfield. M. Object Models: Strategies, Patterns and Applications. Yourdon Press Computing Series. Prentice-Hall, 1997.
- [9] Dardenne A., van Lamsweerde A., and Fickas S. Goal directed requirements acquisition. Science of Computer Programming, (20)1-2, 2003. Pages 350.
- [10] Fickas S. and Helm R. Knowledge Representation and Reasoning in the Design of Composite Systems. IEEE Trans. on Software Engineering, June 1992, pp. 470-482.
- [11] Gamma E., R. Helm, R. Johnson, and J. Vlissides. Design Patterns, elements of Reusable Object-Oriented Software. Addison-Wesley, USA, 1995.
- [12] Giorgini Paolo, Mylopoulos John, Nicchiarelli, and Sebastiani Roberto. Reasoning with Goal Model. In the 21st International Conference on conceptual Modeling (ER2002). Tampere, Finland, October 2002. Springer Verlag.
- [13] Jacobson Ivar, Ericsson Maria and Jacobson Agneta. The Object Advantage: Business Process Reengineering with Object Technology. ACM Press Addison-Wesley Publishing Company, 1995.
- [14] Keller S.E., Kahn L.G. and Panara R.B. Specifying Software Quality Requirements with Metrics. In Tutorial: System and Software Requirements Engineering, R.H. Thayer and M. Dorfman, Eds., IEEE Computer Society Press, 1990.

pp. 145-163.

[15] Lamsweerde A. Goal-Oriented Requirements Engineering: A Guided Tour. Invited minitutorial, Proceeding 5th IEEE International Symposium on (RE'01), Toronto, IEEE, August 2001, pp. 249-263.

[16] Lamsweerde A. van, Letier E. Handling Obstacles in Goal-Oriented Requirements Engineering. IEEE Transactions on Software Engineering, Special Issue on Exception Handling, Vol. 26 No. 10, October 2000, pp. 978-1005.

[17] Letier E. and A. van Lamsweerde. Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering. Proceedings of FSE'04, 12th ACM International Symp. on the Foundations of Software Engineering, Newport Beach (CA), Nov. 2004, pp. 53-62.

[18] Magnus Penker, Hans-Erik Eriksson. Business Modeling With UML: Business Patterns at Work. OMG Press John Wiley & Sons, ISBN: 0-471-29551. USA 2000.

[19] Martínez A., Pastor O., Estrada H. A pattern language to join early and late requirements. In Journal of Computer Science and Technology, special issue on Software Requirements Engineering. (2)5. August 2005, pp. 64-70.

[20] Martínez A., Estrada H. and Pastor O. Generación de Modelos de Requisitos a partir de Modelos Organizacionales: un enfoque basado en Patrones. In Journal of Informatics Technology Management. (2)7, December 2004. ISSN 1657-8236. pp. 11-21.

[21] Martínez A., Castro J., Pastor O., Estrada H. Closing the gap between Organizational Modeling and Information System Modeling. Proceeding of the VI Workshop on Requirements Engineering (WER 2003). Piracicaba SP, Brasil, 2003. pp 93-108.

[22] Pastor Oscar, Gómez Jaime, Infrán E. and Pelechano V. The OO-Method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. In Information Systems 26(7), Elsevier, 2001, pp. 507-534.

[23] Pelechano Ferragud V. Tratamiento de Relaciones Taxonómicas en Entornos de Producción Automática de Software una Aproximación basada en Patrones. Tesis de doctorado del Departamento de Sistemas informáticos y Computación de la Universidad Politécnica de Valencia, España. Valencia 2001.

[24] Potts, C., K. Takahashi and Antón A. I. Inquiry-Based Requirements Analysis. IEEE Software, March 1994. pp. 21-32.

[25] Robinson N. William, Elfson Greg. Goal Directed Analysis with Use Cases. In Journal of Object Technology, (5) 3, 2004, pp. 125-142.

[26] Yu Eric. Towards Modelling and Reasoning support for Early-Phase Requirements Engineering. Proceedings of the 3rd. IEEE International Symposium on Requirements Engineering (RE'97) pp. 226-235. IEEE Computer Society Washington, DC, USA 97.

[27] Yu Eric. Modelling Strategic Relationships for Process Reengineering. PhD Thesis, University of Toronto, Toronto, Canada, 1995.

[28] Yu E., and Mylopoulos J. From E-R to 'A-R' Modelling Strategic Actor Relationships for Business Process Reengineering. International Journal of Intelligent and Cooperative Information Systems, vol. 4, no. 2 & 3, 1995, pp. 125-144.

[1]

