

Generación de Especificaciones de Requisitos de Software a partir de Modelos de Negocios: un enfoque basado en metas ¹

Hugo Estrada^{1,3}, Alicia Martínez^{1,2}, Oscar Pastor¹, Juan Sánchez¹

¹ Universidad Politécnica de Valencia ,
Avenida de los Naranjos s/n , Valencia, España
{hestrada, alimartin, opastor, jsanchez}@dsic.upv.es

² I.T. Zacatepec, Morelos, México

³CENIDET Cuernavaca, Mor. México

Abstract. Los trabajos de investigación actuales en ingeniería de requisitos buscan mecanismos que permitan establecer la relación entre la funcionalidad esperada de un sistema de información y los procesos de negocios a los que éste dará soporte. Este enfoque permitirá asegurar que el sistema de información a desarrollar sea realmente útil en las tareas de los actores organizacionales. Los trabajos de investigación en esta área han determinado que las metas organizacionales son una buena base para establecer la relación entre los objetivos perseguidos por el negocio y los requisitos del sistema de información a desarrollar, ya que todos estos requisitos (funcionales y no funcionales) deben corresponderse con tareas que se desean desempeñar dentro de un proceso de negocios. Los procesos de negocio a su vez, permiten el cumplimiento o satisfacción de alguna o algunas de las metas del negocio. En este trabajo se presenta una propuesta para la obtención de requisitos de software a partir de modelos de negocios. El artículo se divide en dos secciones principales: (a) la construcción de modelos de negocios a partir de un análisis orientado a metas (b) la obtención de un modelo de requisitos de software a partir del modelo de negocios. Este trabajo permite tener un punto de partida sólido para la construcción del sistema de información, donde cada requisito tiene su origen en las metas del negocio.

1 Introducción

El objetivo principal de un sistema de información es automatizar tareas o actividades de un proceso de negocios, permitiendo a los actores organizacionales alcanzar sus metas particulares, así como las metas generales del negocio. Esta es la razón por la que el estudio del ambiente organizacional en el que se implantará el producto software ha sido reconocido como una parte fundamental de la ingeniería de requisitos [2]. En este contexto existen investigaciones que remarcan la importancia de utilizar las metas del negocio para conducir el desarrollo de un producto software.

¹ Este proyecto está parcialmente financiado por la Asociación Nacional de Universidades e Instituciones de Educación Superior ANUIES, México

Los trabajos más significativos en este campo son: a) KAOS [6]: un framework formal basado en lógica temporal y en técnicas de refinamiento de Inteligencia Artificial, donde todos los términos (metas, acciones, estados, etc.) son definidos en forma consistente y rigurosa. El principal énfasis de KAOS es la prueba formal de que los requisitos cumplen las metas que fueron definidas para el sistema de información. b) GBRAM: un Método de Análisis de Requisitos basado en metas [2], a pesar de ser menos formal que KAOS se encuentra más enfocado a la definición de metas y a la unión de éstas con los actores organizacionales.

El objetivo de estos trabajos es ayudar en la definición del conjunto de requisitos necesarios para desarrollar un sistema de información que ayude a los actores organizacionales a lograr sus metas. Estas son algunas de las ventajas de estos enfoques basados en metas:

- Las metas hacen explícita la relación entre las operaciones de bajo nivel del negocio (descritas por los operarios) y las metas de alto nivel (descritas por los administradores de la empresa).
- Las metas permiten determinar en forma precisa la pertinencia de los requisitos. Un requisito es pertinente si es usado para satisfacer por lo menos una de las metas del negocio
- Es posible determinar el conjunto de tareas o actividades necesarias 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 del negocio para determinar la completitud y suficiencia de la especificación de requisitos. La especificación será completa si todas las metas son satisfechas con los requisitos especificados.
- Es posible utilizar las metas para identificar y resolver conflictos entre las diferentes alternativas de realizar una tarea o actividad.

La principal desventaja de las investigaciones antes mencionadas es que sólo consideran la creación de metas del negocio, pero no brindan los mecanismos para derivar un modelo de negocios a partir de estas metas. Esta es una limitación importante debido a que la representación explícita de un modelo de negocios es fundamental para la generación de un modelo de requisitos que proporcione soporte a las tareas expresadas en el modelo de negocios.

El objetivo de este trabajo es obtener requisitos de software a partir de los modelos del negocio. En este trabajo se analizan dos enfoques importantes: el primero es el análisis orientado a metas que sirve como plataforma para construir los modelos de negocios, el segundo enfoque se refiere a la obtención de un modelo de requisitos, representados a través de los casos de uso y de sus correspondientes escenarios, a partir del modelo de negocio. Esta especificación es utilizada posteriormente para la creación semi-automática del sistema de información [8].

El artículo está organizado de la siguiente forma: la sección 2 presenta el marco teórico de este artículo, haciendo énfasis en el modelado de negocio y en modelado de metas. La sección 3 presenta el proceso para elicitación de requisitos organizacionales basado en metas. La sección 4 presenta el proceso de elicitación de requisitos del sistema de software basado en los modelos de negocios. La sección 5 presenta un estudio de los trabajos relacionados en este tema y finalmente, la sección 6 presenta las conclusiones del trabajo.

2 Marco teórico: modelado de negocios y modelado de metas

En esta sección se explican los dos conceptos principales que son utilizados en el proceso de elicitación de requisitos mostrado en este artículo: el modelado de metas y el modelado de negocios. Ambos enfoques se combinan para crear un método de captura de requisitos organizacionales.

2.1 Modelado de Metas

Una meta es un objetivo no operacional. No operacional significa que el objetivo no puede ser formulado en términos de acciones disponibles para algún agente en el sistema, es decir, es una meta que no puede ser satisfecha mediante las transiciones de estado apropiadas de uno de los agentes del negocio.

La captura de las metas puede ser realizada por técnicas de refinamiento o por técnicas de abstracción. A continuación se explica brevemente cada una de ellas.

Identificación de metas por Refinamiento: utiliza un enfoque top-down para la elicitación de metas. Se parte de metas de alto nivel hasta llegar a operaciones del proceso de negocio. En este enfoque se debe responder a la pregunta ¿cómo se logra satisfacer esta meta? Se deben refinar las metas hasta alcanzar metas que puedan ser ejecutadas por agentes individuales, es decir, se trata de metas que pueden ser satisfechas mediante las transiciones de estado apropiadas de uno de los agentes del negocio [6]. En este punto es importante responder la pregunta, ¿existe algún agente que pueda cumplir por sí mismo esta meta mediante la aplicación de sus acciones?

Identificación de metas por Abstracción: utiliza un enfoque bottom-up para la elicitación de metas. Se inicia con las operaciones de los stakeholder o de metas de bajo nivel. Se debe responder a la pregunta ¿cuál es el objetivo de realizar esta tarea? En la mayoría de las ocasiones los actores organizacionales expresan sus requisitos en forma de metas de bajo nivel o de operaciones, esta información es útil para encontrar las metas de alto nivel que las justifican [6].

Las metas que se obtienen como resultado de estos procesos de identificación han sido tradicionalmente utilizadas para determinar, en forma directa, requisitos funcionales y no funcionales del sistema de información. Sin embargo, este enfoque tiene como principal problemática la ausencia de mecanismos que permitan filtrar las tareas o actividades que se desean automatizar con el producto software, debido a que no existe un camino sencillo para insertar el actor sistema de software en la estructura de metas que se genera como resultado del proceso de identificación y derivar hacia este nuevo actor las metas seleccionadas para su automatización. La carencia de una notación gráfica para representar las metas no permite identificar la forma en la cual se relacionan los actores del negocio, cómo dependen entre ellos, que tipo de información se intercambia, ni cómo se afecta el proceso de negocio por el cumplimiento o incumplimiento de las acciones de los actores del negocio.

2.2 Modelado de Negocios

Este trabajo utiliza el framework i* [12] para representar los modelos del negocio. Esta técnica de modelado permite expresar en forma clara las metas de los actores

organizacionales, por lo que es posible determinar cuál es la meta de alto nivel que da sentido a cada una de las tareas del proceso del negocio. El framework i* representa en forma explícita las dependencias entre actores organizacionales, esta información es útil para visualizar en forma gráfica cuáles son las tareas que se esperan que cada actor desempeñe, así como para analizar la repercusión que tiene el cumplimiento o incumplimiento de las tareas asignadas a los actores. El framework i* cuenta con una notación gráfica con pocos elementos, que permite tener una vista unificada del negocio, mostrando actores, dependencias, recursos, tareas, metas en una vista global del proceso de negocio. La mayoría de las técnicas de modelado de negocios utilizan múltiples diagramas (cada una con una vista particular del negocio) para tener la misma expresividad. La notación de i* permite además incluir explícitamente sistemas de software como actores dentro de la organización.

Estas características distinguen al framework i* del resto de técnicas de modelado de negocios [3],[4],[7] que no consiguen expresar en una vista unificada la información relevante.

El framework i* está formado por el modelo de dependencias estratégicas y el modelo de razones estratégicas, que se complementan para proporcionar una visión completa de la naturaleza de la organización.

El *Modelo de Dependencias Estratégicas* modela las dependencias que existen entre los actores de un proceso de negocio. El modelo se representa gráficamente a través de un grafo donde cada nodo representa a un actor o agente, y donde las ligas indican las dependencias que existen entre esos actores para alcanzar sus metas, ejecutar tareas y suministrar o solicitar recursos. El conjunto de nodos y ligas en el modelo de Dependencias Estratégicas forman una red de dependencias, que ayudan a representar gráficamente las relaciones externas entre los actores. El modelo está formado por cuatro tipos de dependencias: en la *Dependencia de Meta*, se depende que un actor realice una actividad, pero sin prescribir la forma en la que debe ser realizada. En una *Dependencia de Recurso*, se depende de que otro actor realice la entrega de un recurso que puede ser material o informacional. En la *Dependencia de Tarea*, se depende de que un actor realice una actividad, pero además se prescribe la forma en que debe ser realizada. La *Dependencia de Meta-suave* es semejante a la dependencia de meta, con la diferencia de que no es posible establecer con precisión el criterio de aceptación de la meta.

El modelo está compuesto además por 3 tipos de actores primitivos: a) el actor dependiente: *dependier*, b) el actor del cual se depende: *dependee* y finalmente c) el objeto alrededor del cual se centra la relación de dependencia: *dependum*.

El *Modelo de Razones Estratégicas* se utiliza para: a) describir los intereses y motivaciones de los participantes (ii) Habilitar la valoración de posibles alternativas en la definición de procesos; (iii) especificar con mayor detalle las razones de la existencia de dependencias entre varios actores. Este modelo extiende al modelo de dependencias estratégicas incorporando dos tipos de constructores: *means-end* cuyo objetivo es representar las diversas alternativas que pueden tomarse para lograr una meta o tarea, y *task-descomposition* que permiten detallar el conjunto de actividades necesarias para lograr un objetivo.

Como se mencionó anteriormente, el framework i* brinda la posibilidad de incluir objetos software como actores de un modelo de negocio, desafortunadamente, actualmente no existe un método bien definido para conducir esta inclusión de actores software. De esta forma, la inserción del actor sistema no está controlada por reglas que aseguren la consistencia del modelo de negocios. Esto puede ocasionar que un

analista diseñe un modelo de negocio demasiado abstracto que no pueda ser posteriormente traducido en especificaciones de requisitos de software.

3. Proceso de Elicitación de requisitos organizacionales basados en metas

Como ya se mencionó anteriormente, uno de los enfoques presentados en este trabajo de investigación es el proceso de elicitación de requisitos organizacionales que tiene como base un análisis orientado a metas. Este método tiene como objetivo capturar la semántica de los procesos de negocios con un método basado en metas, por lo que inicia con la construcción de árbol de refinamiento de metas que permite capturar los objetivos generales de la empresa y a partir de éste se construyen los modelos de dependencias y de razones estratégicas, que conforman el framework i^* , posteriormente se realiza la inserción del actor sistema de software dentro de estos modelos permitiendo así que los modelos de negocios posean una semántica adecuada para derivarse en requisitos de software.

El análisis orientado a metas permita expresar tanto las metas de alto nivel de los administradores como las acciones de bajo nivel de los operarios en una vista unificada y coherente. Este enfoque hace posible determinar cómo cada una de las metas de alto nivel del negocio se implementan en acciones concretas de un proceso de negocios, así como determinar cuál es el objetivo de la ejecución de las tareas de bajo nivel.

El método de elicitación de requisitos organizacionales basado en metas presentado en este trabajo es un método híbrido que combina la identificación de metas por refinamiento y por abstracción, de tal forma que sea posible responder a las preguntas ¿Qué tareas deben ejecutarse para lograr este objetivo? ¿Por qué estas tareas son necesarias? ¿Cómo son ejecutadas las tareas? ¿Quiénes ejecutan las tareas? ¿De quién se depende para su ejecución?

Para ilustrar nuestra propuesta para la creación de modelos de negocios, se utiliza el caso de estudio *Conference Review System*, cuyo propósito es modelar los procesos de negocios relativos a la gestión de los artículos en una conferencia, de tal forma que como resultado final se obtenga la especificación de requisitos de un sistema de software que automatice el proceso de envío, selección y evaluación de artículos para una conferencia. En el caso de estudio se presenta la siguiente casuística: existe un presidente del comité de programa (*PcChair*) que determina los tópicos de interés y selecciona los miembros del comité de programa (*PcMember*). Los autores se encargan de enviar sus artículos. Los miembros del comité de programa se encargan de evaluar los artículos que les han sido asignados y para ello pueden delegar responsabilidades en revisores adicionales. Los revisores son los encargados de evaluar los artículos. Finalmente, el *PcChair* envía las notificaciones de aceptación o rechazo a los autores de artículos.

3.1 Árbol de refinamiento de metas

El primer paso para la elicitación de requisitos organizacionales es la creación de un Árbol de Refinamiento de Metas. La raíz de este árbol representa la meta más general del negocio. Las ramas del árbol representan la agrupación de metas de más bajo nivel para la satisfacción de una meta más general. En nuestra propuesta se ha determinado obtener solo tres niveles de profundidad en cada rama del árbol para una evitar una explosión en el tamaño del modelo de negocios. Finalmente, todas las hojas del árbol representan operaciones.

En este trabajo las metas han sido categorizadas de la siguiente manera:

- **Metas Generales:** son de alto nivel que suelen reflejar el punto de vista que tienen los administradores del negocio. Este tipo de metas no dan origen directo a operaciones, sino a Metas de Logro. Se representan con el símbolo *MG*.
- **Metas de Logro:** son aquellas metas que se subdividen en forma directa en operaciones. Se representan con el símbolo *ML*.
- **Operaciones:** son metas que pueden ser satisfechas mediante las transiciones de estado de uno de los agentes del negocio. Las operaciones representan acciones concretas que se desempeñan para lograr una meta. Puede tratarse de operaciones individuales en las que no se dependa de otro actor, o de operaciones en las que exista dependencias entre actores. En este último caso se trata de operaciones que requieren de información o recursos suministrados por otros actores, y que son indispensables para el cumplimiento de la meta. Se representan con el símbolo *Oper*.
- **Metas en Conflicto:** son metas cuya satisfacción lleva a los actores a estados contradictorios. Se representan con el símbolo *--*.
- **Metas Inclusivas:** son metas que pueden ejecutarse en forma simultánea. Se representan con el símbolo *::*.
- **Metas Exclusivas:** se trata de aquellos casos en los que sólo una de las metas debe de satisfacerse. Se representan con el símbolo *&*.

Para ilustrar nuestro caso de estudio se muestra la descripción textual del Árbol de Refinamiento de Metas (tabla 1). La primera columna muestra en forma jerárquica a las metas identificadas, el orden en el que se encuentra ubicadas indican la precedencia que existe entre ellas, colocándose en primer lugar la meta más general, en el siguiente nivel se encuentran todas las metas de logro que existan, y dentro de cada una de ellas se encontrarán las operaciones necesarias para satisfacer estas metas. La segunda columna indica el tipo de meta u operación. La tercera columna muestra los actores involucrados para alcanzar esa meta. Aquellas operaciones en las que se requiera de la intervención de algún otro actor para lograr la meta, se indican con la aparición de dos actores en la tercera columna del árbol de refinamiento de metas. El primer actor es el actor dependiente y el segundo actor es el actor del cual se depende.

Nombre de la meta	Tipo de meta	Actores
Meta: Realizar el proceso de revisión de artículos.	MG	
1 Enviar Artículo	ML	<i>PcChair</i> - Autor
1.1 Obtener dirección envió	Oper	Revisor- <i>PcChair</i>
1.2 Enviar artículo	Oper	<i>PcChair</i> – Autor
2 Asignar Artículos a PcMember	ML	<i>PcMember</i> – <i>PcChair</i>
2.1 Obtener lista de interés	Oper	<i>PcChair</i> - <i>PcMember</i>

2.2	Generar lista de artículos	Oper	PcChair
2.3	Seleccionar lista de PcMembers	Oper	PcChair
2.4	Identificar y resolver conflictos	Oper	PcChair
2.5	Asignar artículos a PcMembers	Oper	PcMember – PcChair
3	Asignar Artículo a Revisor	ML	Revisor – PcMember
3.1	Obtener Artículos	Oper	PcMember- PcChair
3.2	Asignar artículos a revisores	Oper	Revisor – PcMember
4	Enviar Revisiones	ML	PcChair –PcMember Revisor
4.1	Obtener Artículos	Oper	PcMember- PcChair
4.2	Asignar calificaciones	Oper	PcMember
4.3	Asignar comentarios	Oper	PcMember
4.4	Enviar Revisiones	Oper	PcChair –PcMember/Revisor
5	Enviar Notificaciones	ML	Autor- PcChair
5.1	Obtener revisiones	Oper	PcChair- PcMember/ revisor
5.2	Ordenar artículos	Oper	PcChair
5.3	Resolver casos en conflicto	Oper	PcChair
5.4	Enviar notificación de	Oper	Autor- PcChair
5.5	aceptacion/rechazo		

Tabla 1. Descripción textual del árbol de refinamiento de metas para el caso de estudio Conference Review System

En este trabajo se determinó que todas las metas deberán tener un reflejo en operaciones, de la misma forma que todas las operaciones deberán estar ligadas a metas. Esta restricción impide que puedan especificarse metas que no tengan reflejo en operaciones y operaciones que no satisfagan ninguna meta. La operacionalización de las metas es uno de los pasos fundamentales para la construcción de un modelo de metas que pueda dar origen a un modelo de negocios completo y consistente.

3.2 Creación del Modelo de Dependencias Estratégicas

El Árbol de Refinamiento de Metas será el punto de partida para el proceso de generación de un Modelo de Negocios especificado en el framework i*. El proceso inicia con la creación de un Modelo de Dependencias Estratégicas. A continuación se describen los pasos para construir este modelo:

Paso 1: Utilizar los actores identificados en el Árbol de Refinamiento de Metas (tabla 1 columna 3) para crear los actores del Modelo de Dependencias Estratégicas. Los actores identificados en nuestro caso de estudio son: *PcChair*, *PcMember*, *Revisor* y *Autor*.

Paso 2: Utilizar las Metas de Logro del Árbol de Refinamiento de Metas para crear las dependencias de meta en el Modelo de Dependencias Estratégicas. Para conseguir esto es necesario analizar la descripción de las metas de logro para obtener los elementos de una dependencia de meta en i*: *dependor*, *dependum* y *dependee*. El *dependor* será aquel actor que requiera de la entrega de un recurso, o de la ejecución de una tarea para cumplir alguna de sus metas. El *dependee* es el actor del cual se depende para la ejecución de la tarea o la entrega de un recurso, y finalmente, el *dependum* es el objeto o tarea de cual se depende. En nuestro caso de estudio, por ejemplo, la meta *Asignar artículos a PcMember*, *ML*, *PcMember – PcChair* (Meta 2 del Árbol de Refinamiento de Metas) se corresponde con la dependencia de meta “*Asignar artículos*”, donde el actor *dependee* es el *PcChair*, debido a que es éste el encargado de ejecutar la operación de asignación de artículos. El *PcMember* será el

actor *depender* de la relación de dependencia, debido a que es éste el que depende de que le sean asignados artículos para su revisión. El actor *dependum* de este ejemplo serán los *Artículos* enviados a revisión.

Paso 3: Utilizar las Operaciones del Árbol de Refinamiento de Metas para crear las dependencias de recurso y tarea del Modelo de Dependencias Estratégicas. En el framework *i** no existe el concepto de *operación* por lo que es necesario traducir éstas en dependencias de meta o recurso. Las dependencias de tarea se obtienen de las operaciones donde el actor *depender* prescribe la forma en la que tiene que ser ejecutada la operación. Las dependencias de recurso, en cambio, se presentan en aquellos casos donde el actor *depender* requiere de algún recurso para ejecutar la operación. En nuestro caso de estudio, la Meta de Logro “*Enviar Revisiones*” (tabla 1, meta 4) contiene la operación 4.1 *Obtener Artículos*, en esta operación el actor *PcMember* espera el envío de los artículos por parte del *PcChair*. Esta información es utilizada para crear la dependencia de recurso “artículos”. Por su parte el *PcChair* depende del *PcMember* para el envío de las revisiones de los artículos (tabla 1, operación 4.4), esta operación da origen a la dependencia de recurso: “*revisión*”. Las operaciones que puedan ser desempeñadas por un actor en forma independiente, formarán parte de las acciones internas de cada actor en el modelo de razones estratégicas. La figura 1 muestra la aplicación del proceso de traducción para el caso de estudio analizado.

3.3 Creación del Modelo de Razones Estratégicas

La construcción del Modelo de Razones Estratégicas consiste en definir las operaciones internas que realiza cada actor con el fin de alcanzar satisfactoriamente las dependencias que tiene con los actores con los cuales está relacionado. Esta información es también obtenida del Árbol de Refinamiento de Metas.

El proceso consiste en seleccionar cada actor del modelo de negocios y colocar sus Metas de Logro como metas internas. Estas metas deberán ligarse con las operaciones o metas necesarias para su cumplimiento o satisfacción. El objetivo de esta etapa es construir un árbol de refinamiento de metas internas en cada uno de los actores del modelo de negocios, utilizando tanto las operaciones en las que exista dependencia de otros actores como aquellas que pueden ser logradas por algún actor en forma independiente. De esta forma, algunas de estas operaciones internas estarán ligadas a las dependencias de tarea o de recurso definidas en el modelo de dependencias estratégicas.

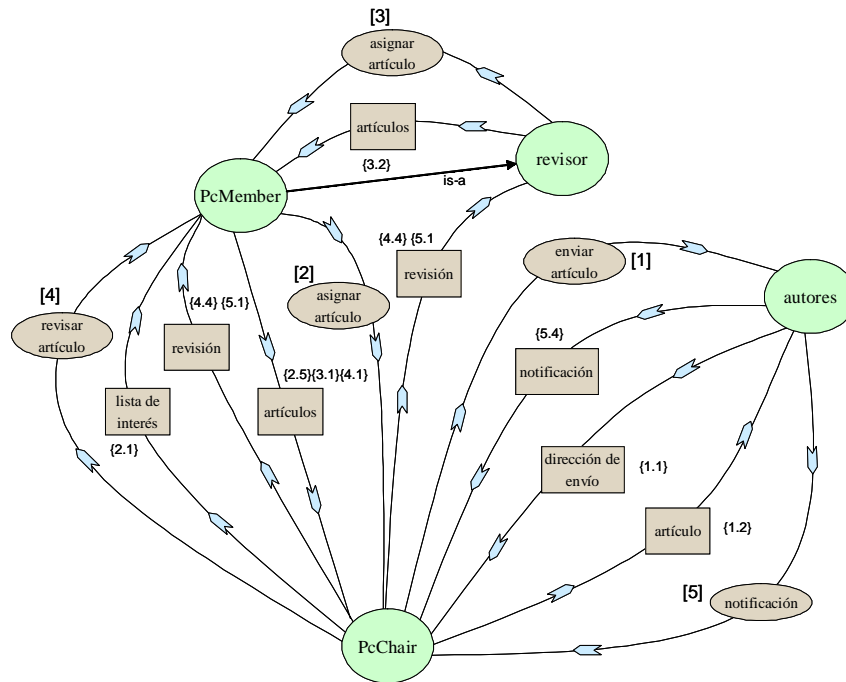


Figura 1 Modelo de Dependencias Estratégicas para el caso de estudio

Por ejemplo, en el caso de la Meta de Logro *Asignar Artículos a PcMember* (tabla 1, meta 2), el actor que actúa como *dependee* en esta dependencia (*PcChair*) tendrá una tarea con el mismo nombre que representa la raíz de un árbol de refinamiento de metas internas al actor. Posteriormente se crean las tareas: *obtener lista de interés*, *identificar conflictos*, *generar lista de PcMember*, *generar lista de artículos* y *asignar artículo*, y se unen a la tarea raíz antes mencionada. Estas tareas representan las acciones internas que el *PcChair* debe ejecutar para satisfacer la dependencia que tiene con el *PcMember*. Como ya se ha comentado estas operaciones internas pueden estar ligadas relaciones de dependencia entre actores, de esta forma, por ejemplo, la dependencia de recurso "lista de interés" estará ligada a la tarea *obtener lista de interés* del actor *PcChair*, así como a la tarea *enviar lista de interés* del actor *PcMember* (ver figura 2).

4. Proceso de Elicitación de requisitos de software basado en un modelo de negocios

Existen antecedentes de investigaciones para generar modelos de casos de uso a partir de modelos de negocios en los trabajos de Santander y Castro [10]. Sin embargo, en estos trabajos no se presenta un proceso para la creación de los modelos de negocios, por lo cual, la complejidad de realizar un modelo de casos de uso sólo se ve desplazada a la complejidad de creación de un modelo de negocios. En los trabajos de Santander y Castro se propone la dedicación de un mayor esfuerzo en la etapa de

transformación del modelo de negocios al modelo de casos de uso. Este enfoque tiene como problemática principal la ausencia de guías que permitan dirigir la construcción del modelo de negocios, sin estas guías, los ingenieros de requisitos pueden crear modelos de negocios demasiados abstractos, complejos o ambiguos, y que por lo tanto su traducción a un modelo de requisitos del sistema no pueda ser guiada por un enfoque metodológico. Esto ocasiona que se requieran demasiadas decisiones por parte del ingeniero de requisitos, y que por consiguiente el proceso de transformación se vuelva sumamente complejo y consuma demasiado tiempo y esfuerzo.

En este trabajo se propone un mayor análisis en la creación del modelo de negocios, proporcionándose guías que permitan restringir el tipo y la complejidad de los modelos de negocios resultantes.

De esta forma es posible asegurar que los modelos de negocios resultantes estén correctamente adaptados para la generación de casos de uso del sistema de software. Por lo tanto antes de iniciar con el proceso de elicitación de requisitos de software, proponemos añadir al actor sistema dentro del esquema del modelo de negocios, el propósito de aislar las operaciones del negocio que se desean ejecutar automáticamente por el sistema de información. La subsección 4.1 detalla este proceso de inserción del actor sistema, mientras que la subsección 4.2 detalla el proceso de generación del modelo de requisitos utilizando los modelos de negocios que contienen al actor sistema.

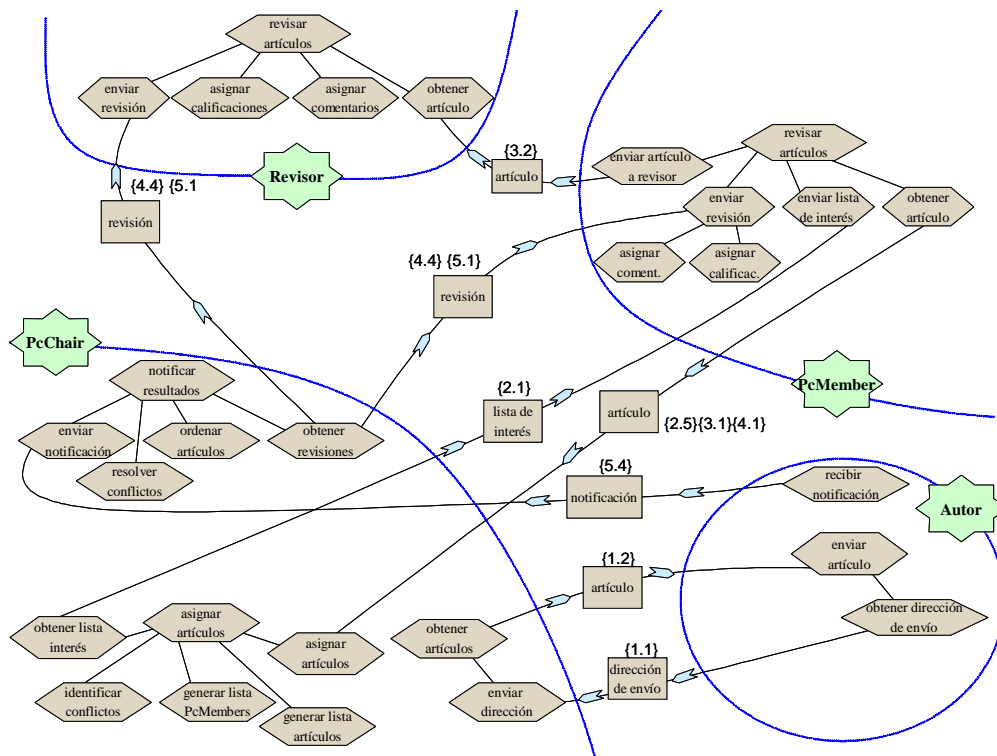


Figura 2 Modelo de Razones Estratégicas del caso de estudio Conference Review System.

4.1 proceso de inserción del actor sistema de software

El proceso de inserción del actor sistema pretende orientar al ingeniero de software a la captura de los requisitos de software utilizando los modelos de negocio. La estrategia consiste en determinar con el tipo de interacción que tiene cada uno de los actores que componen los modelos de negocios con el sistema de software.

Un concepto importante utilizado en este proceso es el de *módulo*. Un módulo representa un árbol de refinamiento de metas internas en un actor, que es representado como descomposiciones de tareas en el modelo de razones estratégicas del framework i*. Un actor puede tener más de un módulo, esto indica que el actor debe cumplir más de una meta en el modelo de negocios. Por ejemplo, en la figura 2 es posible identificar que *PcChair* tiene los módulos *Notificar resultados* y *Asignar Artículos*. A continuación se presentan una serie de guías que permiten insertar al actor sistema:

Guía 1. Insertar el actor sistema dentro del modelo de negocios, identificando además aquellos módulos que requieran ser automatizados por el sistema de información.

Guía 2. Mover los módulos a automatizar desde los actores organizacionales hacia el actor sistema. Sólo será pertinente mover un módulo desde un actor organizacional hacia el actor sistema cuando la tarea principal (raíz del módulo) necesite ser automatizada. Para mover los módulos es necesario:

1. Dejar una copia de la raíz del módulo dentro del actor organizacional.
2. Crear una dependencia de tarea con el mismo nombre del módulo entre el actor organizacional y el sistema, esta dependencia indicará que el actor organizacional depende del sistema para ejecutar esa tarea.

Es posible que existan módulos en los que algunas de las tareas deban realizarse en forma manual y donde el sistema de información sólo sea utilizado para introducir o recibir datos, en este caso es necesario que estas tareas permanezcan ligadas al módulo del actor organizacional. De esta forma al final de esta tarea existirán dos módulos con el mismo nombre, uno colocado en el actor organizacional con las operaciones manuales y otro en el actor sistema con las operaciones automatizadas.

Guía 3. Existen tareas que al ser trasladadas hacia módulos en el actor sistema requieren de algún tipo de información proveniente del actor organizacional para poder ejecutarse. En este caso será necesario crear nuevas dependencias de recurso entre el actor sistema y el actor organizacional. Este es el caso de la tarea *Generar lista de PcMember* que originalmente se encontraba en el actor *PcChair* y que al ser trasladada hacia el actor sistema requiere de la intervención de *PcChair*, esto es debido a que el sistema no puede determinar por si solo una lista de *PcMembers*, en este caso es necesario crear la dependencia de recurso *Lista de PcMembers* entre el actor sistema y el actor *PcChair*.

Guía 4. En caso de actividades en las que se presenten dependencias de recurso, es necesario crear nuevas tareas de envío y recepción de recursos ligadas a la meta principal del módulo, En esta propuesta se recomienda crear únicamente dependencias de tarea y de recurso entre los actores organizacionales y el actor sistema, de esta forma se evita colocar dependencias de meta que tendrían que ser posteriormente derivadas en metas de recurso o tarea.

La figura 3 muestra el resultado de la aplicación de estas guías para nuestro caso de estudio.

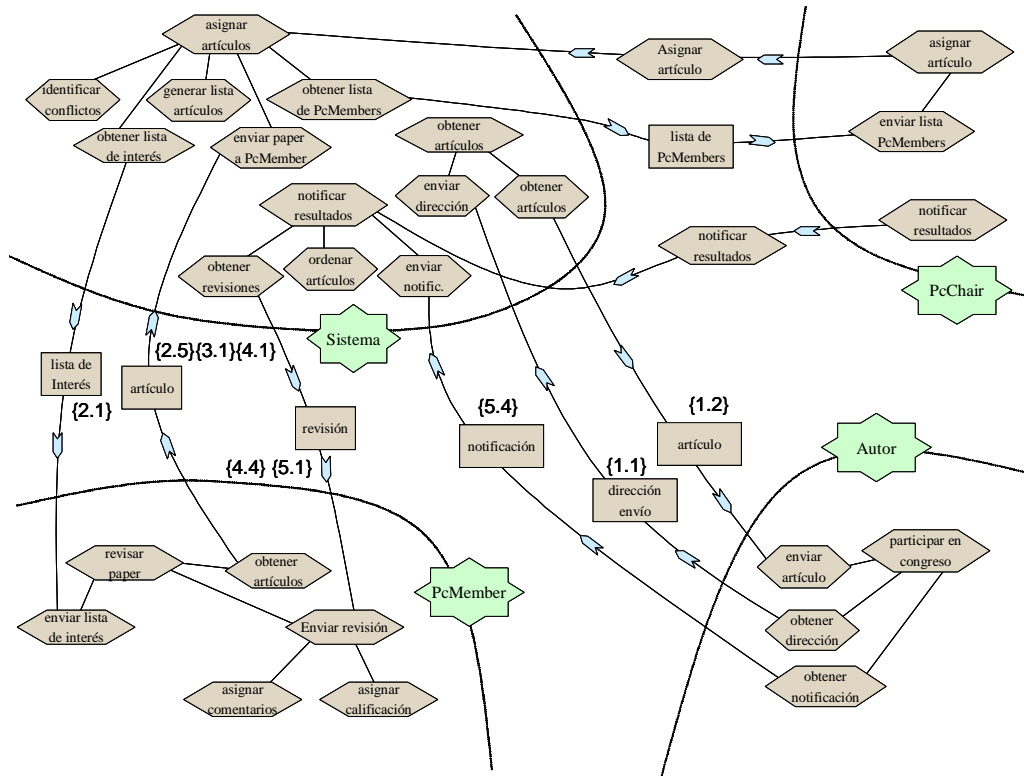


Figura 3 Inserción del actor sistema de software en el modelo de negocios

La aplicación de estas guías permitirá que estos modelos de negocios puedan ser traducidos en forma natural a un modelo de requisitos basados en casos de uso y escenarios, los cuales pueden ser traducidos a su correspondiente modelo conceptual y en forma posterior a un modelo de ejecución [8]. Es necesario hacer notar que este enfoque no ha sido utilizado por ninguna otra propuesta para traducir modelos de negocios en especificaciones software.

4.2 Generación de casos de uso y escenarios a partir de modelos de negocios

Para guiar el proceso de mapeo entre los modelos de negocios y un modelo de casos de uso UML se definieron un conjunto de pasos que establecen la correspondencia entre los elementos del modelo de negocio especificado en el framework i* (que tiene integrado en forma explícita al actor sistema) y un modelo de casos de uso y sus correspondientes escenarios representados en una plantilla.

Paso 1. Obteniendo un modelo de Casos de Uso.

El primer paso de este proceso es la determinación de un conjunto de dependencias de meta relevantes que serán utilizadas para generar los casos de uso. Estas dependencias relevantes son aquellas dependencias de meta que fueron obtenidas del conjunto de Metas de Logro de Árbol de Refinamiento de Metas, y que en forma posterior fueron derivadas en relaciones de dependencia de recurso o tarea entre algún

actor organizacional y el actor sistema. Por ejemplo, la dependencia de Meta *Enviar Revisión* (meta 4 figura 1) entre los actores: *PcChair* y *PcMember* fue derivada en la dependencias de recurso: *Obtener Artículo* (dependencia 4.1 figura 3) y *Enviar Revisión* (dependencia 4.4 figura 3) entre el actor sistema y el *PcMember*, por lo que se considera que debe existir un caso de uso que tenga como nombre *Enviar Revisión*. Aquellas relaciones de dependencia de meta originadas de las Metas de Logro de Árbol de Refinamiento de Metas que no fueron posteriormente derivadas en dependencias hacia el actor sistema, representan actividades manuales en las que no existe interacción entre el usuario y el sistema de software, de tal forma que estas actividades no necesitan ser modeladas como casos de uso.

Las dependencias de recurso y tarea por sí mismas no originan casos de uso, ya que éstas deben formar parte de alguna dependencia de meta más general.

Las dependencias de meta suave no fueron analizadas ya que estas se enfocan a requisitos no funcionales no tratados en este trabajo.

Paso 2. Descubriendo actores de los casos de uso:

Para determinar a los actores de los casos de uso, es necesario analizar aquellas dependencias de recurso y tarea derivadas a partir de una dependencia de meta relevante para aparecer como un caso de uso. En estas dependencias se tiene siempre al actor sistema de software en un extremo de la dependencia, y por el otro a un actor organizacional que aparece como usuario del sistema, y que por tanto debe aparecer como actor del caso de uso. Por ejemplo, el actor *PcChair* se relaciona con el actor sistema para el cumplimiento de dependencias derivadas de la meta *Asignar artículo*, por lo que *PcChair* aparecerá como el actor primario del caso de uso.

Por otro parte, es necesario determinar la existencia de actores que comparten un comportamiento común, ya que esto puede originar relaciones de herencia entre actores. Por ejemplo, el actor *PcMember* contiene toda la funcionalidad de un *Revisor* (revisar artículos y entregar revisiones) además de su propio comportamiento (asignar revisores e identificar áreas de interés), por lo que podemos establecer que el *PcMember* es una generalización del actor *Revisor*.

Aquellos actores que no posean relaciones de dependencia con el actor sistema no deberán ser considerados actores potenciales de un caso de uso. En la figura 4 se muestra el resultado de la aplicación de estos pasos a nuestro caso de estudio.

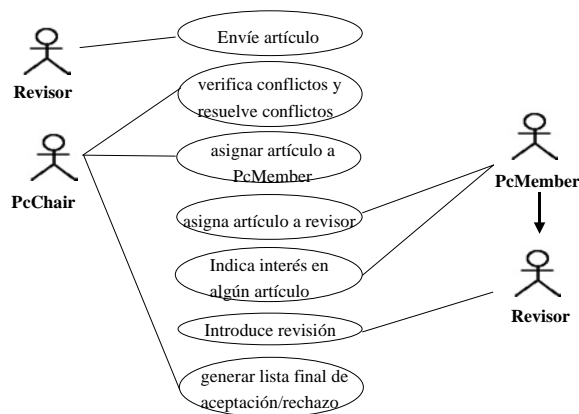


Figura. 4 Modelo inicial de casos de uso para el sistema Conference Review System

El modelo de casos de uso que se propone está basado en UML y contiene como componentes básicos los actores y casos de uso. El modelo posee una representación gráfica que muestra las relaciones entre estos dos componentes básicos, así como un modelo de descripción que contiene la secuencia de eventos que intercambian los actores y el sistema cuando se ejecuta un caso de uso.

Los actores representan las entidades externas que intercambian información con el sistema, mientras que los casos de uso representan la funcionalidad del sistema que se ofrece al exterior.

Paso 3. Representación de escenarios

Para cada dependencia de meta elegida como un caso de uso es necesario crear un modelo de descripción, el cual utiliza plantillas para mostrar la secuencia de eventos que intercambian los actores y el sistema. Para esto se utiliza una variante de la plantilla propuesta por L. Constantine [5] (para detalles puede consultarse Sánchez [11]). En la plantilla es necesario detallar tanto las intenciones de los usuarios, como las responsabilidades del sistema, esta división permite detectar aquellos casos donde los actores solicitan servicios, además de aquellos donde el sistema actúa como suministrador de información. Esta información se obtiene del Modelo de Razones Estratégicas utilizando los pasos mostrados a continuación. Para ejemplificar el uso de los pasos, se ha elegido el caso de uso *Enviar Revisiones* (dependencia 4 figura 1), para el cual se genera la plantilla mostrada en figura 5.

- **Paso 3.1** El proceso de construcción del modelo de descripción inicia con la elección de alguna de las metas relevantes seleccionadas como un caso de uso (heurística 1), así como del conjunto de dependencias de tarea y recurso que se derivan de ésta.
- **Paso 3.2** El nombre del caso de uso en el modelo de descripción se obtiene directamente del nombre de la dependencia de meta elegida. En este ejemplo, una de las dependencias de meta entre el *PcMember* y el *PcChair* se denomina *Enviar Revisiones* nombre que se le asigna también al caso de uso.
- **Paso 3.3** El nombre del actor primario del caso de uso se obtiene del actor que actúa como actor *dependee* en la dependencia de meta que da origen al caso de uso, en este ejemplo, el actor que participa como *dependee* es el actor *PcMember*
- **Paso 3.4** Para determinar las intenciones de los usuarios con respecto al actor sistema, es necesario delimitar cuales son aquellas relaciones de recurso, o meta que están relacionadas con la meta elegida como caso de uso. Una vez delimitadas estas dependencias es necesario identificar aquellas en las que el actor Sistema se encuentra como *dependee*, es decir, aquellas en las que el sistema espera acciones o recursos del usuario. Siguiendo el ejemplo, las intenciones de los usuarios en la dependencia *Enviar Revisiones* son: La selección de la opción Introducir Revisión, la introducción de los datos de identificación del artículo, y la introducción de la evaluación.
- **Paso 3.5** Para determinar las responsabilidades del sistema es necesario identificar las relaciones de recurso o meta en las que el actor Sistema actúa como *dependee*, es decir, aquellas en las que los actores esperan acciones o recurso del Sistema. En el ejemplo, las responsabilidades del sistema son: la solicitud de revisiones a los *PcMembers*, y el envío de los artículos para ser revisados. En la plantilla fueron adicionadas responsabilidades del sistema

que tienen como objetivo la validación de los datos introducidos por el usuario.

Use Case Name: Enviar Revisiones	
Relations	
Include:	None
Extend:	None
Description	
Preconditions	El revisor se ha identificado en el sistema
Postconditions	None
Primary Actor	PcMember
Secondary Actors	
Event Flow	
User intentions	System responsibilities
1. El PCMember selecciona "Introducir Revisión"	
	2. El sistema solicita el artículo a revisar
3. El revisor introduce el código de artículo	
	4. El sistema comprueba que el artículo existe y que está asignado al revisor
	5. El sistema pide datos de la revisión
6. El revisor introduce los valores de evaluación del artículo y selecciona "almacenar"	
	7. El sistema pide los comentarios del artículo
8. El PCMember introduce los comentarios y selecciona "almacenar"	
	9. El sistema almacena la información de la revisión
Asynchronous extensions	
El Revisor Puede seleccionar Stop en cualquier momento	
Synchronous extensions	
Si en el punto 4 el artículo no está asignado al revisor el sistema emite un mensaje de error	

Figura. 5 Especificación del caso de uso *Enviar Revisiones*

5. Trabajos relacionados

Las primeras aproximaciones de modelado de negocios aparecieron con Bubenko [3], quien hace énfasis en la necesidad de tomar en cuenta el contexto organizacional durante el proceso de especificación de requisitos, aunque sus modelos eran informales y no permitían un trabajo claro de sistematización y automatización.

Posteriormente aparecieron trabajos de investigación que pretendían producir especificaciones de software partiendo de modelo organizacionales con es el caso del Multi-Formalismo para el Modelado Formal de Requisitos para Sistemas de Manufactura [9]. Esta técnica de modelado usaba en forma simultanea varios

lenguajes complementarios (Albert II y CIMOSA) y reusaba patrones ya predefinidos para el modelado de sistemas de manufactura.

Recientemente tenemos los trabajos de Alencar [1], en los que se proponen guías para la traducción de una especificación temprana de requisitos (utilizando para esta etapa los modelos estratégicos del framework i*), en una especificación tardía de requisitos, (utilizando la notación de pUML con anotaciones en OCL). Esta investigación dio un nuevo enfoque a los trabajos de modelado de negocios, abriendo la posibilidad de generar la definición de clases un sistema de información a partir de las primitivas de un modelo de procesos de negocio. Este enfoque tiene como principal debilidad la falta de un modelo intermedio entre el modelo de negocios y el modelo de requisitos del sistema, que permita determinar la correspondencia entre las tareas y actividades del negocio y la funcionalidad esperada por parte de los usuarios del sistema de software, que generalmente representa sólo un subconjunto de estas tareas y actividades organizacionales.

Los trabajos de Santander y Castro [10] proponen una serie de guías empíricas para traducir un modelo de negocios en un modelo de casos de uso. Sin embargo, como se comentó con anterioridad, estos trabajos tienen su principal debilidad en la falta de un proceso que guíe la construcción de modelos de negocios que tengan el nivel de abstracción y complejidad necesarios para ser traducidos en especificaciones de un modelo de requisitos.

Conclusiones

En este trabajo se ha presentado un proceso para la generación de Modelos de Negocios a partir de Metas. En dicho proceso se han definido un conjunto de pasos que permiten generar modelos de negocios que reflejen con precisión las metas que cada actor persigue para cumplir, a su vez, con las metas generales del negocio. Se presentan también los pasos necesarios para integrar el actor sistema de software dentro del modelo de negocios y para derivar hacia este actor la funcionalidad que se desea automatizar en el modelo de negocios.

Se han aplicado las guías para el modelado organizacional a un caso de estudio particular, mostrándose la representación gráfica de cada uno de los modelos de negocios generados.

Finalmente, se presentan los pasos necesarios para traducir el modelo de negocios en una especificación de casos de uso compatible con UML, y sus respectivos escenarios donde se describe la funcionalidad esperada del sistema de software.

El modelo de casos de uso generado sirve como entrada a un proceso semiautomático que genera, por una parte, una especificación del comportamiento del sistema, y por otra, prototipos de interfaces de usuario [8]. La especificación se emplea en la última fase del método, para animar las interfaces generadas.

References

1. Alencar, F., Castro, J., Cysneiros, G., Mylopoulos, J., "From Early Requirements Modeled by i* Technique to Later Requirements Modeled in Precise UML", In

- Proceedings of the "III Workshop on Requirements Engineering", pp 92-108, Rio de Janeiro, (2000).
2. Anton Annie, "Goal Based Requirements Analysis," in Proc. Second Int. Conference on Requirements Engineering. ICRE '96, pp. 136-144, 1996.
 3. Bubenko, J. A., jr and M. Kirikova, (1994), "Worlds" in Requirements Acquisition and Modelling, 4th European - Japanese Seminar on Information Modelling and Knowledge Bases, Kista, Sweden, H. Kangassalo and B. Wangler (Ed.), IOS, The Netherlands.
 4. Cesare S. Mark Lycett, "Business Modelling with UML, distilling directions for future research", Proceedings of the Information Systems Analysis and Specification (ICEIS) 2002.
 5. Constantine L.L; Lockwood L.A.D. "Software for Use: A practical Guide to the Models and Methods of Usage-Centered Design". Addison Wesley 1999.
 6. Dardenne, A. van Lamsweerde and S. Fickas, "Goal Directed Requirements Acquisition," Science of Computer Programming, vol. 20, pp. 3-50, Apr. 1993.
 7. Loucopoulos Pericles, Evangelioa Kayakli: Enterprise Modelling and the Teleological Approach to Requirements Engineering, International Journal of Cooperative Information Systems (IJCIS), 1995.
 8. Martinez Alicia, Hugo Estrada, Juan Sánchez, From Early Requirements to User Interface Prototyping: A methodological approach, Submitted to 17th IEEE International Conference on Automated Software Engineering 2002, September 23-27, 2002, Edinburgh, UK
 9. Petit M, ""Formal Requirements Engineering of Manufacturing Systems: A Multi-Formalism and Component-Based Approach. Computer Science Department, University of Namur, Belgium, PhD. Thesis, 1999
 10. Santander, V. F. A., Castro, J. B., "Developing Use Cases from Organizational Modeling", IV Workshop on Requirements Engineering (WER) , Buenos Aires Argentina, 2001.
 11. Sánchez J; Pastor O; Fons J.J. "From User Requirements to User Interfaces: a methodological approach". CAISE 2001. 4-8 Junio 2001. Interlaken Suiza.
 12. Yu, Eric, Modelling Strategic Relationships for Process Reengineering, Phd Thesis, University of Toronto, (1995).