

Generación de Modelos de Requisitos a partir de Modelos Organizacionales: Un enfoque basado en patrones



ALICIA MARTÍNEZ REBOLLAR

Docente en el Instituto Tecnológico de Zacatepec, México

Estudiante de Doctorado en la Universidad Politécnica de Valencia, España

alimartin@dsic.upv.es

PAIS (MÉXICO-ESPAÑA)

HUGO ESTRADA ESQUIVEL

Docente e Investigador del Centro Nal. De Investigación y Desarrollo Tecnológico, México.

Estudiante de Doctorado en la Universidad Politécnica de Valencia, España

hestrada@dsic.upv.es

PAIS (MÉXICO-ESPAÑA)

OSCAR PASTOR LÒPEZ

Director y Docente en el Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, España

opastor@dsic.upv.es

PAIS (ESPAÑA)

RESUMEN.

Actualmente, la fase temprana de la Ingeniería de Requisitos, en la cual se aborda el estudio de la organización en la cual se implantará el sistema de software, es un área incipiente de investigación para múltiples grupos de investigación en el mundo. En esta área se utilizan modelos que permiten describir una organización utilizando actores, metas, procesos organizacionales y relaciones entre actores. Por otra parte, la fase tardía de la Ingeniería de Requisitos (Requisitos tardíos), donde se analiza la funcionalidad esperada del sistema de información, cuenta actualmente con un nivel mayor de madurez, ya que existen múltiples técnicas y herramientas para describir el sistema de software que se va a desarrollar, dentro de su ambiente operacional, junto con sus funciones y cualidades relevantes. Sin embargo, a pesar de que existen metodologías que dan soporte en forma separada a estas dos fases de la Ingeniería de Requisitos, se ha descuidado el desarrollo de técnicas que permitan derivar, en forma metodológica un modelo de requisitos tardíos a partir de los requisitos tempranos. Esto es debido, en gran medida a la gran diferencia entre los niveles de abstracción que existen entre estas dos especificaciones. Por lo tanto el objetivo de este trabajo es proporcionar reglas sistemáticas para generar requisitos de sistemas de información a partir de la información relevante de un modelo organizacional.

PALABRAS CLAVES

Modelado organizacional, Modelo de requisitos, Casos de uso, Lenguajes de patrones.

1. INTRODUCCIÓN

La ingeniería de requisitos es actualmente una de las áreas más incipientes en el campo de investigación de la ingeniería de software. Derivado del enorme interés y la gran cantidad de trabajos en esta línea, ha sido necesario clasificar los trabajos de la ingeniería de requisitos en dos grandes líneas: requisitos tempranos y requisitos tardíos. En este sentido, actualmente existen múltiples trabajos de investigación cuyo objetivo es representar en forma precisa un modelo organizacional (requisitos tempranos) [5][10][3][6]. En estos

trabajos se proponen primitivas conceptuales para modelar los aspectos relevantes en el contexto organizacional: actores, metas organizacionales, relaciones entre actores. De forma similar, existen múltiples trabajos de investigación que abordan el desarrollo de modelos de requisitos (requisitos tardíos) que reflejan en forma precisa la funcionalidad esperada del sistema de información [17][7][12][9][19]. Sin embargo, aunque la fase temprana y tardía de la ingeniería de requisitos, han sido estudiadas en forma aislada, actualmente, no existe una solución definitiva al problema de ligar, en forma metodológica los modelos organizacionales y de requisitos. Esto es debido, en gran medida a la gran diferencia entre los niveles de abstracción que existen entre

estas dos especificaciones. Esta carencia de métodos de trazabilidad entre ambos modelos ha afectado la aplicación práctica de las técnicas de modelado organizacional en ambientes integrados de producción de software. En este sentido, uno de los requisitos básicos para asegurar la utilidad de cualquier propuesta que pretenda incluir una fase de modelado organizacional en un proceso de producción de software es la determinación de una propuesta metodológica que permita utilizar los elementos de un modelo organizacional para obtener las funciones esperadas del sistema de información.

Actualmente, la mayoría de los procesos de producción de software que integran una etapa de ingeniería de requisitos lo hacen implementando herramientas de modelado de casos de uso. Los casos de uso están solamente equipados para responder al "qué" acciones ejecutará el sistema pero no pueden dar una respuesta al "por qué" deben ser construidos. De esta forma, los enfoques orientados a la implementación no consideran la información sobre el ambiente organizacional que envolverá al sistema de información, y de cómo el sistema de información puede ser el eje en torno al cual se desarrollen los procesos organizacionales de la empresa.

En este sentido, en un proceso de producción de software que no tenga como primera etapa el modelado de procesos organizacionales, cualquier esfuerzo para generar un prototipo del sistema de información puede verse disminuido por la incapacidad de asegurar, de antemano, la utilidad del sistema en el contexto de las tareas organizacionales.

La motivación principal de la investigación presentada en este artículo es proporcionar reglas sistemáticas que permitan obtener el modelo requisitos a partir de un modelo organizacional. Esto se logra reduciendo el nivel de abstracción entre los requisitos tempranos y los requisitos tardíos, para lo cual se propone un lenguaje de patrones que ayude al analista a realizar este proceso de forma sistemática.

Este artículo está estructurado de la siguiente forma: en la sección 2 se discuten las metodologías utilizadas: modelado organizacional, lenguaje de patrones e ingeniería de requisitos. En la sección 3 se presenta una descripción de la propuesta presentada. La sección 4 muestra el modelo organizacional inicial que se utiliza como base para el proceso de generación de los requisitos del sistema, además se expone el caso de estudio utilizado en este artículo. La sección 5 muestra el lenguaje de patrones que se propone para reducir el nivel de abstracción entre las fases de tempranas y tardías de los requisitos. La sección 6 muestra el método de generación del modelo de requisitos a partir de un modelo organizacional. Finalmente la sección 7 presenta las conclusiones de la propuesta presentada.

2. METODOLOGÍAS UTILIZADAS

En esta sección se presentan las metodologías que dan soporte al trabajo de investigación presentado en este artículo. En primer lugar se presenta el Framework Tropos, el cual da soporte en las tareas de modelado organizacional. Posteriormente, se presenta una visión general acerca de los Patrones de Software y por último se presenta un breve

resumen de la Ingeniería de Requisitos.

2.1 FRAMEWORK TROPOS

La metodología de Modelado Organizacional Tropos [5] es actualmente una de las técnicas más conocidas y referenciadas en el área de modelado organizacional orientado al desarrollo de sistemas de software. Tropos ha adoptado los conceptos ofrecidos por el Framework i*[20], para soportar la etapa de modelado y análisis de requisitos tempranos. Este Framework está orientado a describir en forma gráfica, la intencionalidad que se presenta en los actores organizacionales. Para esto, Tropos propone dos modelos: el modelo de dependencias estratégicas y el modelo de razones estratégicas. Este último modelo, que es utilizado en este artículo, permite representar a una organización como una red de actores que interactúan entre ellos para alcanzar sus metas particulares, así como las metas del negocio. Este modelo describe un razonamiento más profundo de las motivaciones que existen detrás de cada una de las relaciones de dependencias entre los actores organizacionales.

Las dependencias expresan relaciones intencionales que existen entre los actores organizacionales con el objeto de cumplir sus objetivos estratégicos. Una dependencia describe un acuerdo entre dos actores, el dependiente (el actor dependiente) y el dependee (el actor del cual se depende). El tipo de la dependencia describe la naturaleza del acuerdo: a) las dependencias de meta son usadas para representar la delegación de responsabilidades para alcanzar una meta. b) Las dependencias de tarea representan situaciones donde el dependee es requerido para desempeñar una actividad. c) las dependencias de recurso indican los casos donde se requiere al dependee para proporcionar un recurso. El modelo de razones estratégicas proporciona además, elementos de estructuración de las metas y tareas internas de los actores organizacionales. La liga de descomposición de tareas permite representar la combinación de tareas necesarias para lograr una meta, mientras que la liga de medios fines permite representar las diversas alternativas que pueden ser tomadas para satisfacer una meta o una tarea.

2.2 LENGUAJE DE PATRONES

Un patrón es una descripción de una solución común a un problema recurrente, que puede ser aplicado a un contexto determinado [8]. Existen diversos tipos de patrones, tales como Patrones de arquitectura [4], que muestran las arquitecturas de alto nivel de un sistema, Patrones de diseño [15][16], los cuales están orientados hacia la programación, o Patrones que están enfocados a la administración de proyectos [1].

En este artículo se propone un lenguaje de patrones organizacionales que permiten reducir el nivel de abstracción del modelo organizacional para hacerlo más próximo al modelo de requisitos. Esto se logra insertando el actor sistema de software (SSA) en el modelo organizacional y redirigiendo hacia este actor las relaciones de dependencias que se presentaban entre los actores organizacionales.

De esta forma, los patrones propuestos permiten analizar los elementos organizacionales, tales como; tareas, recursos y metas, con el fin de introducir un nuevo actor que represente

el sistema de software que se pretende construir, y el cual se debe apegar a los objetivos y dependencias de tareas y recursos que existen entre los actores organizacionales.

2.3 INGENIERÍA DE REQUISITOS

En las últimas décadas se ha venido reconociendo de manera amplia que los requisitos del software deben ser gestionados adecuadamente para el éxito de los proyectos de desarrollo de software en general [13]. Numerosas investigaciones han demostrado la importancia de los requisitos de software con respecto a la calidad [18], disminución de fallos [19], y coste de corrección de errores [11] La respuesta que ha surgido de la ingeniería de software ante la relevancia de requisitos ha sido fomentar la creación de la disciplina de la Ingeniería de Requisitos.

Siguiendo con esta disciplina surge el modelado de requisitos que pretende representar adecuadamente los requisitos de software. Uno de los párrafos más citados en la bibliografía de la Ingeniería de Software dice: "La parte más difícil de construir un sistema es precisamente qué construir. Ninguna parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con gente, máquinas y otros sistemas. Ninguna parte del trabajo afecta tanto al sistema si es hecha mal. Ninguna es tan difícil de corregir más adelante. Entonces la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto" [2].

Los casos de uso son un método que, justamente, ayudan al ingeniero de software a llevar adelante esta parte del desarrollo de un sistema de software. Las técnicas basadas en escenarios han sido utilizadas en la ingeniería de software para entender los modelos y los requisitos de los usuarios.

Algunos enfoques propuestos para el uso de escenarios para elicitar y validar requisitos pueden analizarse en las siguientes referencias, [18][9][14].

3. DESCRIPCIÓN DE LA PROPUESTA

En esta sección se presenta una vista general del método que se propone para obtener un modelo de requisitos a partir de un modelo organizacional.

En la figura 1 se muestran las 3 fases de las que consta el método propuesto.

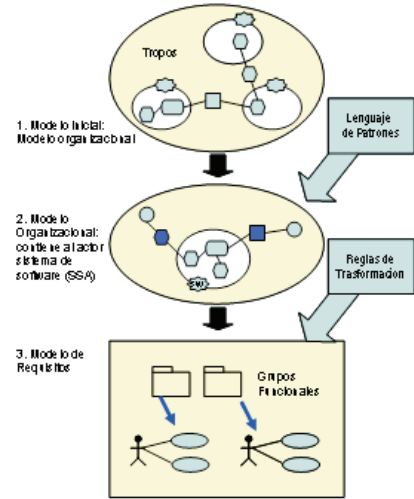


Figura 1. Representación esquemática del método propuesto

La fase 1 se inicia con la especificación de un modelo organizacional, el cual debe estar representado en el

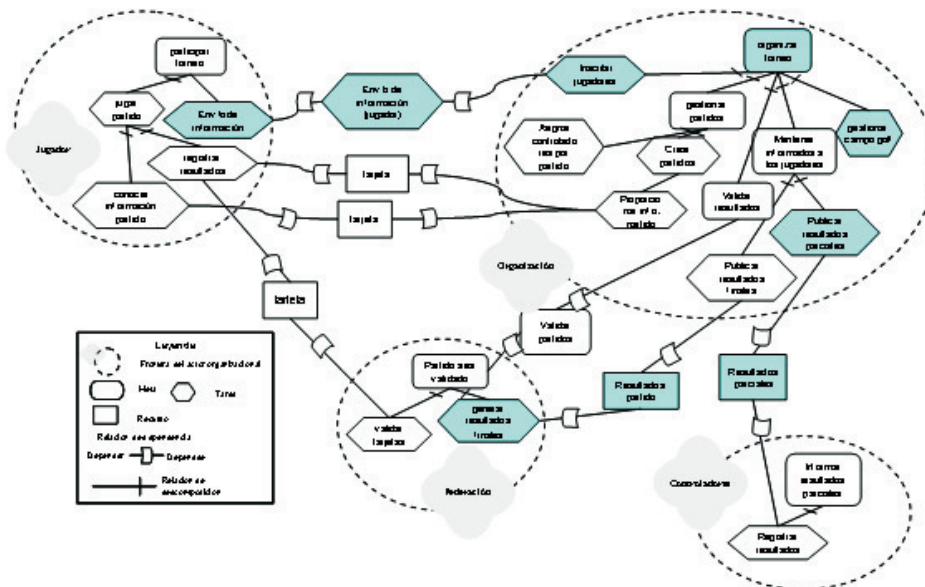


Figura 2 Modelo Inicial del caso de estudio: gestión y seguimientos de torneos de golf

framework Tropos. La fase 2, se obtiene al utilizar un lenguaje de patrones, denominado FELRE (From Early Requirements to Late Requirements) que también ha sido propuesto, con el fin de generar un nuevo modelo organizacional que lleve implícito al actor sistema de software, el cual lo hemos denominado SS-BM (Software System-Business Model). Esta fase permitirá reducir el nivel de abstracción entre un modelo organizacional "puro" y un modelo de requisitos. Por último la fase 3 permite derivar un modelo de requisitos a partir de un modelo organizacional de una forma sistemática, por lo que se proponen guías para la realización de este proceso.

Caso de Estudio. Con el objeto de ilustrar nuestra propuesta, se ha utilizado el caso de estudio: "Gestión y seguimientos de Torneos de Golf". El objetivo de este caso de estudio es modelar los procesos organizacionales de una empresa que se dedica a la gestión y administración de Torneos de Golf. Los Torneos de Golf, al ser validados por la Federación de Golf, son puntuables para el ranking de un jugador en el Campeonato de Golf. Por su parte, una de las preocupaciones principales de la empresa es proporcionar resultados parciales de cada partido, por lo que, existen Controladores que registran los resultados de los jugadores para unos hoyos dados. A continuación, por consideraciones de espacio se muestra un fragmento del modelo organizacional, del caso de estudio analizado en este artículo (figura 2). En este modelo es posible observar que los actores involucrados en este proceso organizacional son: la Federación de Golf, la Organización, los Jugadores y los Controladores. En este proceso del negocio, la Organización depende de que la Federación valide los resultados de cada partido. A su vez, la Federación depende que los jugadores entreguen las tarjetas con los resultados finales de cada partido. Los jugadores, a su vez, dependen de que la Organización le haga entrega de las tarjetas para anotar los resultados de cada hoyo.

Los elementos que aparecen remarcados en la figura 2, serán utilizados para ejemplificar la inserción del actor sistema del nuevo modelo organizacional en la siguiente sección.

4. GENERACIÓN DE SS-BM UTILIZANDO UN LENGUAJE DE PATRONES

Los lenguajes de Patrones tienen actualmente una amplia aplicación como herramientas para la definición de marcos metodológicos que guíen los procesos de modelado y diseño. Siguiendo esta línea de trabajo, se ha desarrollado un Lenguaje de Patrones (FELRE) para ayudar a reducir el nivel de abstracción que existe entre los requisitos tempranos y los requisitos tardíos.

El nuevo modelo organizacional (SS-BM) generado por la aplicación de los patrones, tiene como objetivo redirigir al actor sistema de software (SSA) las tareas del negocio que se pretenden automatizar, así como redirigir las relaciones de dependencias que origine introducir al SSA en el SS-BM.

4.1 IMPLEMENTACIÓN DEL LENGUAJE DE PATRONES (FELRE)

El lenguaje de patrones propuesto está compuesto por cinco patrones que permiten guiar de forma sistemática al analista para obtener el nuevo SS-BM. A continuación se describe brevemente el proceso de implementar el lenguaje de

patrones.

Paso 1. Identificar las tareas que se pretenden automatizar. En la figura 2, se muestran remarcados los elementos que se pretenden automatizar y que nos permitirán ejemplificar cada uno de los patrones propuestos. Las tareas y metas remarcadas han sido seleccionadas por la empresa que ha proporcionado el caso de estudio analizado en este artículo.

Paso 2. Colocar el SSA en el SS-BM, así como los actores, que en el modelo inicial, tengan alguna tarea, meta o relación de dependencia que se pretenda automatizar utilizando el sistema de información. En el caso de estudio analizado, los actores con esta característica son: los Jugadores, la Organización, la Federación y los Controladores.

Paso 3. Trasladar o redirigir las tareas o metas que se pretenden automatizar hacia el SSA. Para este paso, es necesario realizar lo siguiente:

Paso 3.1 Realizar un recorrido infijo en cada uno de los actores organizacionales del modelo inicial. Un actor puede estar compuesto de una o varias metas o tareas, las cuales pueden a su vez subdividirse en otras metas o tareas. De esta forma, estas subdivisiones llegan a formar una estructura arborescente. En la figura 3 se muestra un ejemplo de este recorrido. En nuestro caso de estudio "Gestión y seguimientos de Torneos de Golf", si realizamos el recorrido infijo del actor Organización de la figura 2, la primera tarea analizada sería Inscribir jugadores, posteriormente se analizaría la tarea Organizar Torneo, seguido de la tarea Asignar controladores por partido, y así sucesivamente hasta terminar de recorrer el árbol de tareas internas del actor Organización.

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, se deben analizar todos los elementos de esa relación, es decir, el depender, el dependee y el dependum. Este análisis permite identificar los patrones de automatización que se presentan en el modelo organizacional. Los patrones pueden ser: Patrón Automatización Tarea Final sin dependencias, Patrón Automatización Tarea/Meta general, Patrón Automatización Tarea depender-dependee. Patrón Automatización Tarea dependee.

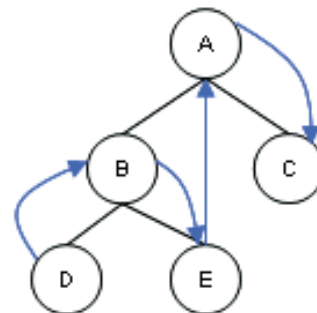


Figura 3. Ejemplo de recorrido infijo

Continuando con el ejemplo del paso 3, para satisfacer la tarea inscribir jugadores, el actor depender (Organización) depende

que el actor dependee (Jugador) realice la tarea envío de información de sus datos. En este caso, tanto el proceso de inscribir jugadores como el de enviar la información del jugador ha sido seleccionados para ser automatizados, por lo que el patrón identificado en este proceso es: Patrón Automatización Tarea depender-dependee.

Paso 3.3 Utilizar el patrón adecuado para trasladar los elementos que se desean automatizar al SSA. Una vez identificado el patrón que se requiere utilizar, se siguen los pasos descritos en ese patrón.

4.2 DESCRIPCIÓN DE LOS PATRONES DE AUTOMATIZACIÓN

En esta sección se explican brevemente cada uno de los patrones del lenguaje de patrones FELRE.

1) Patrón de automatización de tarea final sin dependencias

Contexto:

Existen diferentes tipos de tareas en una empresa, uno de estos tipos es la tarea final, que es aquella tarea que no se descompone en otras metas o en otras tareas. Las tareas finales sin dependencias, son aquellas tareas finales que no requiere de la intervención de otro actor para desarrollarla, solamente necesitan del actor que la contiene, es decir, no tienen asociada ninguna dependencia con otro actor.

Problema: Cuando se modela un negocio, en donde se pretende construir un sistema de software, se requiere analizar las tareas que deberán automatizarse y cómo afectarán a los actores y a las dependencias que existen entre dichos actores. Si la tarea que se requiere automatizar es una tarea final, la cual no tiene asociada ninguna dependencia entonces se deben analizar otras situaciones. Las fuerzas asociadas a este problema son:

- Que una tarea final requiera automatizarse.
- Que la tarea final requiera automatizarse y que necesite del actor que la desarrollaba anteriormente para terminar de desarrollarla completamente.
- Que la tarea final requiera automatizarse y que necesite de otros actores para terminar de desarrollarla además del actor que la contiene.

Solución: Cuando se pretende automatizar una tarea final sin dependencias, el primer paso a realizar es trasladar esta tarea al SSA del SS-BM. El siguiente paso consiste en analizar si se requiere de la intervención del actor que contenía originalmente la tarea para terminar de desarrollar la tarea, o si en cambio, el SSA puede ejecutar la tarea en forma independiente. En el caso de que se requiera la intervención de otro actor, ya sea para desarrollar la tarea o para obtener algún recurso, se deben crear nuevas relaciones de dependencia entre estos actores. Estas nuevas dependencias pueden ser de: tarea o de recurso las cuales indicarán ya sea la introducción de información o el desempeño de alguna actividad.

Ejemplo: Este tipo de patrón lo encontramos al realizar el recorrido infijo del actor Organización (ver figura 2). La tarea Gestionar campo de gol, la cual es una tarea final sin dependencias, cumple con las características de este patrón. Aplicando el patrón, esta tarea se traslada al SSA y se genera

una nueva tarea de dependencia entre el actor Organización y el SSA, que indica que el primer actor le proporcionará la información de los campos de golf, por lo cual campos de golf, se pone como un parámetro de la dependencia de tarea. El resultado de este proceso puede observarse en la figura 4.

2) Patrón de Automatización de Tarea o Meta General

Contexto: Los elementos del Modelo de Razones Estratégicas del Framework Tropos son: la descomposición de tareas y las ligas de medios fines. La descomposición de tareas indica que una tarea principal (a la que se le ha denominado tarea general ó nodo padre) necesita ser ejecutada completando la totalidad de las subtareas (denominadas nodos hijos) en las que ha sido refinada. Por otro lado, en la liga de medios fines, se requiere que sólo alguna de las tareas en las que se subdivide una tarea general sea ejecutada para que la tarea padre pueda ser también satisfecha.

Problema: Cuando una Tarea o Meta general se analiza para una posible automatización, se debe tener en cuenta sus nodos hijos asociados, porque el traslado de esta tarea o meta al SSA dependerá de la decisión de automatizar algunos de sus nodos hijos. A continuación se explican las fuerzas asociadas a este problema.

- Que al menos un nodo hijo requiera automatizarse.
- Que la Tarea o Meta general posea una dependencia con otro actor organizacional.

Solución: Para determinar si la Tarea o Meta general debe trasladarse al SSA, se requiere analizar primero a sus nodos hijos, si al menos uno de ellos se automatiza entonces será necesario trasladar la tarea o meta general al SSA, para realizar esto, el primer paso es trasladar la tarea o meta general al SSA del SS-BM, el segundo paso es asociarle las tareas o submetas que dependían de ella y que han sido trasladadas al SSA. Si la tarea o meta general tiene asociada una dependencia con otro acto es necesario hacer un análisis utilizando los patrones, Automatización Tareas DependierDependee, Automatización Tarea Dependier, o Automatización Tarea Dependee según sea el caso.

Ejemplo: Este tipo de patrón los encontramos al realizar el recorrido infijo del actor Organización (ver figura 2).

La meta Organizar Torneo cumple con las características de este patrón. Para determinar si esta meta será trasladada al SSA, se analizan los nodos hijos que tenga asociados. En este caso, algunos de sus nodos ya han sido trasladados al SSA, (las tareas: inscribir jugadores, publicar resultados parciales y gestionar campos de golf), por lo que esta meta se trasladará hacia el SSA. Para analizar el resultado de la aplicación de este patrón observar la figura 4.

3) Patrón de Automatización de las Tareas Dependier-Dependee

Contexto: En los modelos del framework i* los actores organizacionales se relacionan entre sí a través de dependencias. Cada dependencia está compuesta por [20]: La dirección de la dependencia, que establece quien es el actor

dependen y el actor dependee y El tipo de la dependencia (recurso, meta, meta-suave o recurso).

Problema: Uno de los problemas fundamentales al insertar el SSA en el modelo organizacional es el traslado de las tareas a automatizar al SSA. Si la tarea que se está analizando tiene asociada una relación de dependencia, es necesario analizar todos los elementos de esa relación: depender, dependee y dependum. Cuando se desea automatizar tanto la tarea del actor depender como la tarea del actor dependee, el objeto dependum es el que dará la pauta de los pasos a seguir para trasladar estas tareas al SSA. Las fuerzas asociadas a este problema son las siguientes:

Si el dependum es un recurso, en este caso se debe analizar si se requiere automatizar la generación del recurso, la obtención del recurso o ambas. Generalmente, en un modelo de razones estratégicas, en una relación de dependencia de recurso, uno de los actores involucrados tiene la tarea interna de proporcionar el recurso y el otro actor tiene la tarea de obtener el recurso.

Si el dependum es una tarea o meta, entonces se debe analizar si la automatización de la tarea por parte del dependee también permitirá que el depender pueda también realizar la tarea u obtener los resultados de la tarea realizada por el dependee. Generalmente en un modelo de razones estratégicas, en una relación de dependencia de tarea, el actor depender depende que el actor dependee realice una tarea interna para la cumplir el objetivo de una tarea o de una meta.

Solución: Cuando los dos actores involucrados en la relación de dependencia intentan automatizar sus tareas, se debe analizar al objeto dependum para seguir las instrucciones dependiendo el caso que sea:

a) Si el dependum es un recurso, se analizan las tareas o metas a las que se encuentra ligadas el recurso, realizando los siguientes pasos:

Paso 1. Se traslada la tarea de generación (del actor dependee) u obtención (del actor depender) del recurso (cualquiera que sea el caso) al SSA. En caso de que no esté especificada explícitamente la tarea de generación u obtención, en lugar de trasladar la tarea del depender o dependee, se puede crear una nueva tarea interna en el SSA.

Paso 2. La dependencia de recurso se redirecciona entre el depender o dependee y el SSA.

Paso 3. Al realizar el redireccionamiento del paso anterior, la dependencia original entre los actores organizacionales desaparece, por lo que uno de los actores organizacionales, ya sea el actor depender o dependee (esto depende de si se ha elegido automatizar la tarea de recepción o generación del recurso), han quedado sin relaciones de dependencia con el sistema, ni con el otro actor organizacional involucrado en la dependencia. En este caso, es necesario crear una relación de interacción entre este actor que ha quedado sin dependencia y el actor sistema de software. Una relación de interacción está representada gráficamente por una flecha entre dos actores

organizacionales y representa el hecho de que un actor puede activar una tarea del SSA.

Paso 4. Las relaciones de dependencia que se generen de la automatización de la obtención o generación del recurso deberán marcarse para indicar que estas relaciones de dependencia se encuentran asociadas.

Paso 5. Las tareas internas de los actores organizacionales que fueron trasladadas hacia el SSA, se marcarán para que se identifique que ya han sido analizadas y se pueda continuar con el análisis de la inserción del SSA, en el SS-BM.

b) Si el dependum es una tarea, entonces se debe analizar si la automatización de la tarea por parte del dependee permitirá que el depender pueda también realizar la tarea u obtener los resultados de la tarea realizada por el dependee. Una vez entendido lo que se pretende automatizar, se analizan las tareas o metas a las que se encuentra ligadas la tarea, realizando los siguientes pasos:

Paso 1. La tarea del actor depender se traslada al SSA.

Paso 2. Se crea una relación de dependencia, que puede ser: recurso, tarea o simplemente de interacción entre el actor depender y el SSA. El actor dependiente será el SSA. Esta nueva relación de dependencia indica la introducción o recepción de información por parte del actor organizacional con el SSA.

Paso 3. La tarea del actor dependee (de la relación de dependencia analizada originalmente) se traslada al SSA, esta tarea se colocará como una descomposición de tarea que estará ligada a la tarea del depender, colocada en el paso 1.

Paso 4. Se crea una relación de dependencia, que puede ser: recurso, tarea o simplemente de interacción entre el actor dependee y el SSA. El actor dependiente será el SSA. Esta nueva relación de dependencia indica la introducción o recepción de información por parte del actor organizacional con el SSA.

Paso 5. Las relaciones de dependencia que se generen de la automatización de las tareas deberán marcarse para indicar estas relaciones se encuentran asociadas.

Paso 6. Las tareas internas de los actores organizacionales que fueron trasladadas hacia el SSA, se marcarán para que se identifique que ya han sido analizadas y se pueda continuar con el análisis de inserción del SSA.

Ejemplo: Este tipo de patrón los encontramos al realizar el recorrido infijo del actor Organización (ver figura 2). La tarea Inscribir jugadores cumple con las características de este patrón, debido a que tiene asociada la dependencia de tarea Envío de información (jugador) que también se desea automatizar. Una vez aplicados los pasos del patrón para trasladar las tareas que se desean automatizar, se produjeron los siguientes pasos, que es posible observar en la figura 4: se crea una descomposición de tarea, donde la tarea general es inscribir jugadores y se descompone en la tarea obtención de información, además se generan una relación de dependencia y una relación de interacción, las cuales han sido marcadas con el número 1.

4) Patrón de Automatización de la Tarea Dependee

Contexto: En una relación de dependencia entre dos actores, uno de ellos actúa como actor dependiente (depende), es decir depende de otro actor para alcanzar sus metas, cumplir sus tareas u obtener algún recurso. El actor depende será por tanto el actor del cual depende para satisfacer la relación de dependencia [20].

Problema: Como se ha comentado anteriormente, uno de los problemas fundamentales al insertar el SSA y la determinación de las tareas a automatizar. Si la tarea que se está analizando tiene asociada una relación de dependencia, será necesario analizar los elementos que componen esta dependencia: depende, depende y dependum. Cuando sólo se desea automatizar la tarea del actor depende de una relación de dependencia, será necesario analizar el dependum de la relación de dependencia. Este elemento será el que dé la pauta de los pasos a seguir para trasladar estas tareas al SSA. Las fuerzas asociadas a este problema son las siguientes:

- Que la tarea que requiera automatizarse esté ligada a un recurso.
- Que la tarea que requiera automatizarse esté ligada a otra tarea.

Solución: En el caso de que sólo se desee automatizar la tarea del actor depende, se debe analizar al objeto dependum al cual está ligado la tarea del actor depende. Existen dos posibilidades para analizar:

a) Si el dependum es un recurso, se realizan los siguientes pasos:

Paso 1. Se traslada la tarea a automatizar al SSA del SS-BM.

Paso 2. Se analiza si es necesaria la intervención de actor que contenía la tarea (actor depende de la relación de dependencia original) para que la tarea pueda ser ejecutada de forma automática por el SSA. En caso de requerirse la intervención del actor, se crea una nueva dependencia de tarea o de recurso entre este actor y el SSA. El actor depende de esta nueva relación de dependencia será el SSA.

Paso 3. La dependencia de recurso se coloca igual entre los actores organizacionales del SS-BM.

b) Si el dependum es una tarea, se realizan los siguientes pasos:

Paso 1. Se traslada la tarea a automatizar al SSA del SS-BM.

Paso 2. Se analiza si es necesaria la intervención de actor que contenía la tarea (actor depende de la relación de dependencia original) para que la tarea pueda ser ejecutada de forma automática por el SSA. En caso de requerirse la intervención del actor, se crea una nueva dependencia de tarea o de recurso entre este actor y el SSA. El actor depende de esta nueva relación de dependencia será el SSA.

Paso 3. La dependencia de tarea se redirecciona, colocando como actor depende al SSA y como actor depende al actor que actuaba también como depende de la relación de dependencia original.

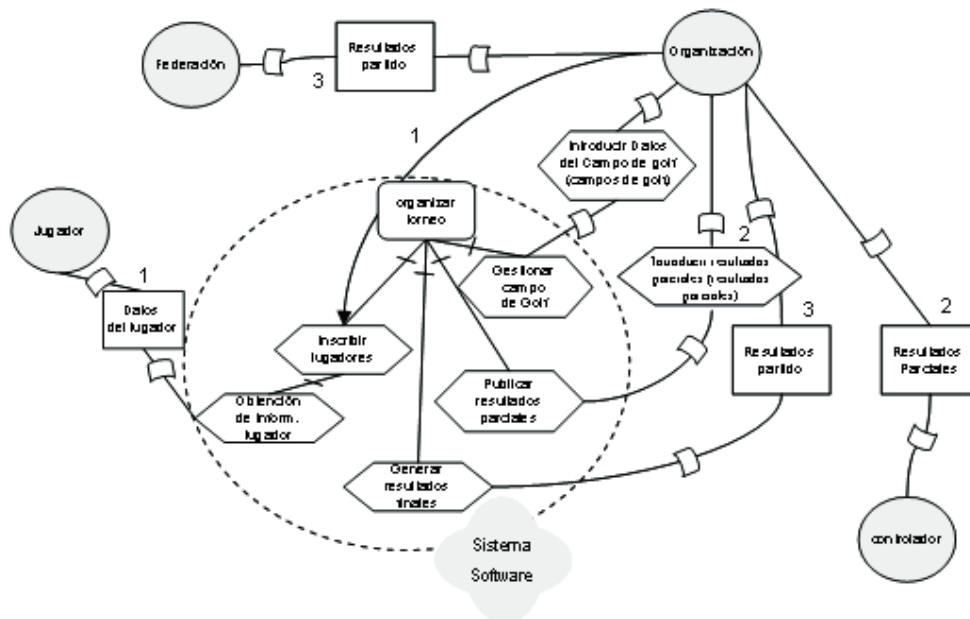


Figura 4. SS-BM del caso de estudio: gestión y seguimientos de torneos de golf

Ejemplo: Este tipo de patrón lo encontramos al realizar el recorrido infijo del actor Organización (ver figura 2). La tarea Publicar resultados parciales cumple con las características de este patrón, debido a que se desea automatizar la tarea del depender únicamente. En este caso se siguen los pasos a aplicar cuando el dependum de la relación de dependencia es un recurso. El resultado de la aplicación de este patrón se puede observar en la figura 4. Las dependencias que se generan o que se modifican, se encuentran marcadas con el número 2.

5) Patrón de Automatización de la Tarea Dependee

Contexto: En una relación de dependencia entre dos actores, uno de ellos actúa como dependee. Este actor será el responsable de satisfacer la dependencia, y por tanto, será el actor del cual se depende para conseguir un objetivo o tarea organizacional [20].

Problema: La inserción del SSA en el modelo organizacional implicar realizar un análisis profundo para determinar las tareas que es necesario trasladar desde los actores organizacionales al SSA. Usualmente, las tareas internas de los actores organizacionales están ligadas a relaciones de dependencia con otros actores del negocio. En este caso, será necesario analizar los elementos de la dependencia. Cuando sólo se desea automatizar la tarea del actor dependee de una relación de dependencia, será necesario analizar el dependum de la relación de dependencia. Este elemento será el que dé la pauta de los pasos a seguir para trasladar estas tareas al SSA. Las fuerzas asociadas a este problema son las siguientes:

- Que la tarea que requiera automatizarse esté ligada a un recurso.
- Que la tarea que requiera automatizarse esté ligada a otra tarea.

Solución: Cuando sólo se requiera automatizar la tarea del actor dependee, se debe analizar al objeto dependum al cual está ligado la tarea del actor depender. Existen dos posibilidades para analizar:

a) Si el dependum es un recurso, se realizan los siguientes pasos:

Paso 1. Se traslada la tarea a automatizar al SSA del SS-BM.

Paso 2. En este paso es necesario analizar si el actor depender podrá acceder directamente al sistema para enviar el recurso, o si en cambio, el recurso será enviado al actor que posee la dependencia.

Paso 2.1 Si el depender no tiene acceso al sistema, la dependencia de recurso se queda igual entre los actores organizacionales del SS-BM. Se crea otra dependencia de recurso entre el SSA y el depender (de la relación de dependencia original), donde el depender de esta nueva relación de dependencia será el SSA.

Paso 2.2 Si el actor depender puede tener acceso al sistema,

la dependencia de recurso se coloca del depender hacia el SSA. Se crea una relación de interacción entre el actor que actuaba como dependee y el SSA.

b) Si el dependum es una tarea, se realizan los siguientes pasos:

Paso 1. Se traslada la tarea a automatizar al SSA del SS-BM.

Paso 2. Se redirecciona la dependencia de tarea entre el actor que actúa como depender y el SSA.

Paso 3. Se analiza si para el desarrollo de esta tarea de forma automática requiere de la intervención del actor que la contenía, en este caso se coloca una nueva dependencia de tarea o de recurso, entre estos dos actores. La dependencia la tendrá el SSA. O puede colocarse una relación de interacción entre estos actores.

Ejemplo: Este tipo de patrón lo encontramos al realizar el recorrido infijo del actor Federación (ver figura 2). La tarea Generar resultados finales cumple con las características de este patrón, debido a que se desea automatizar la tarea del actor dependee únicamente. En este caso se siguen los pasos a aplicar cuando el dependum se refiere a un recurso. El resultado de la aplicación de este patrón se puede observar en la figura 4. Las dependencias que se generan o que se modifican, se encuentran marcadas con el número 3.

5. GENERACIÓN DEL MODELO DE REQUISITOS

La obtención del modelo de requisitos a partir de un modelo organizacional tiene como punto de inicio el SS-BM generado por la aplicación de los patrones de traducción. A continuación se muestran las guías que permiten llevar a cabo este proceso de forma sistemática.

Guía 1. Descubriendo Grupos funcionales

Los grupos funcionales permiten describir los distintos subsistemas en los que se dividirá el sistema de información. Cada uno de los grupos funcionales hace referencia a un elemento manipulable, a través de interacciones por los usuarios del sistema. En esta propuesta, estos elementos se corresponden con los recursos que pasan de los actores organizacionales al SSA. Estos recursos deben ser gestionados por el sistema, por lo que cada recurso dará origen a un grupo funcional.

Para descubrir los grupos funcionales del modelo de requisitos, es necesario analizar cada una de las dependencias que existan entre los actores organizacionales y el SSA: a) En caso de se trate de una dependencia de recurso, se generará un grupo funcional para gestionar este recurso. b) En caso de trate de una dependencia de tarea, se debe analizar si esta tarea genera o modifica un recurso, en este caso, deberá generarse también un grupo funcional. Se debe asegurar que no se generen grupos funcionales repetidos. El nombre del grupo funcional puede componerse de la palabra "gestión de" y el nombre del recurso modificado, si en el sistema existen varios recursos iguales el nombre aparecerá en plural.

En la Figura 5 se muestra un fragmento del SS-BM, de la

dependencia de tarea "Introducir resultados parciales" entre el SSA y el actor Organización. El recurso asociado a esta dependencia es "resultado parcial", como esta tarea será desarrollada más de una vez por el actor organización, el nombre del grupo funcional estará en plural. Por lo tanto el grupo funcional generado será "gestión de resultados parciales".

Guía 1.1 Definición de los casos de uso por defecto de los grupos funcionales. Cada grupo funcional tendrá por defecto los casos de uso creación, eliminación y modificación, los cuales permitirán la manipulación del recurso por parte del sistema de software. En la Figura 6 se muestran los casos de uso para el grupo funcional Gestión de Resultados parciales.

A continuación se descubrirán los actores de los casos de uso, así como el resto de los casos de uso para cada grupo funcional.

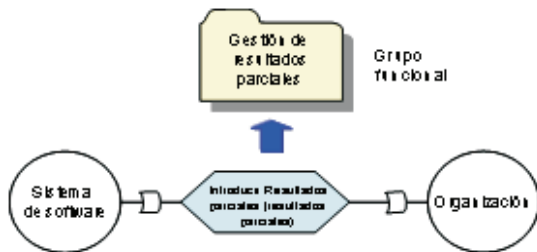


Figura 5. Grupo funcional: Gestión de Resultados parciales

Guía 1.2 Descubriendo los actores de los casos de uso de un grupo funcional. Los actores de los casos de uso serán aquellos actores organizacionales que interactúen con el SSA. La interacción puede ser vista a través de una relación de dependencia o a través de una relación de interacción. Para determinar los actores de los casos de uso de cada uno de los grupos funcionales se debe analizar nuevamente la dependencia que dio origen al grupo funcional, tomando en cuenta lo siguiente:

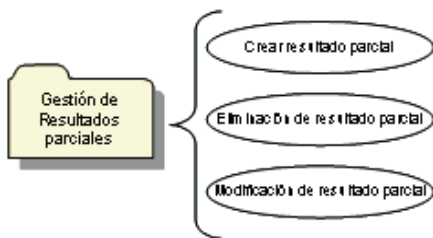


Figura 6. Casos de uso de un grupo funcional

a) **Si se trata de una dependencia de tarea o de recurso,** el actor que no sea el sistema de software se considerará como el actor de los casos de uso del grupo funcional de esa dependencia (ver Figura 7).



Figura 7. Descubriendo el actor de los casos de uso a partir de una dependencia

b) **Si existen varias dependencias asociadas.** Generalmente, la satisfacción de un objetivo organizacional no se consigue con una única dependencia entre dos actores. En la mayoría de los casos es necesario definir un conjunto de dependencias relacionadas entre si para la satisfacción de un objetivo común. Estas dependencias pueden presentarse entre distintos actores organizacionales. A este conjunto de dependencias se les ha denominado, en este trabajo, dependencias asociadas. Las dependencias asociadas pueden ser de meta, recurso, tarea o simplemente pueden ser una relación de interacción. En la figura 4 se muestran varias dependencias asociadas, cada grupo de dependencias asociadas esta marcado con un mismo número.

En caso de existir dependencias asociadas, los actores de los casos de uso serán los mismos que aquellos que tengan una dependencia con el actor sistema de software. Esto indica que estos actores tendrán acceso a los casos de uso del grupo funcional. Por ejemplo, la dependencia de recurso Datos del jugador, entre el Jugador y el SSA, tiene la relación de interacción asociada, marcada con el número 1(ver figura 8). Los actores de los casos de uso identificados en el grupo funcional Gestión de jugadores serán el Jugador y la Organización.

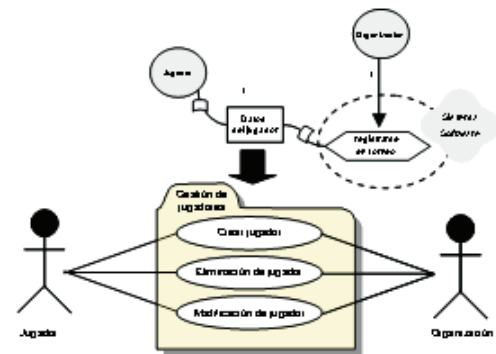


Figura 8. Descubriendo los actores de los casos de uso a partir de relaciones asociadas

Guía 2. Descubriendo los casos de uso de las tareas del actor sistema de software. La estrategia de este paso consiste en analizar cada una de las tareas internas en el SSA buscando casos de uso candidatos. Para este análisis se siguen los siguientes pasos:

Paso 1. Analizar cada una de las metas o tareas generales que se encuentren en el SSA, siguiendo un recorrido infijo para el análisis de los elementos que componen cada meta o tarea general. Este proceso será similar al que se utilizó en la inserción del SSA.

Paso 2. Analizar cada tarea o meta identificada si puede ser considerada por sí misma un caso de uso. Una tarea o meta será candidata a ser un caso de uso si ésta tiene una relación de interacción o si tiene asociada alguna dependencia con algún actor organizacional, esto se debe a lo que describe un caso de uso, que son las interacciones que realiza un actor con el sistema para alcanzar una meta [7].

Paso 3 Determinar el grupo funcional al que pertenece el caso de uso identificado, esto se logra tomando en cuenta lo siguiente.

Paso 3.1 Si la meta o tarea analizada se encuentra ligada a una relación de dependencia, en este caso, el grupo funcional del caso de uso identificado será el grupo funcional que se haya generado de esa dependencia.

Por ejemplo, en la Figura 9, la tarea analizada es Publicar resultados parciales, esta tarea según el paso 2, es considerada un caso de uso, por lo tanto para identificar el grupo funcional al que pertenece es necesario analizar la dependencia a la que se encuentra asociada. El grupo funcional que generó esta dependencia fue: gestión de resultados parciales, por lo tanto, el grupo funcional del caso de uso publicar resultados parciales será este grupo funcional.

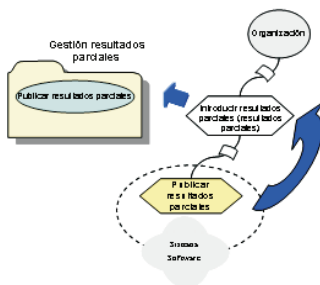


Figura 9. Determinando el grupo funcional de un caso de uso a partir de una dependencia

Paso 3.2 Si la meta o tarea analizada se encuentra ligada a una relación de interacción, en este caso, será necesario analizar el o los recursos que modifica esta tarea. En caso de que aun no se haya definido un grupo funcional de este recurso, se generará en este momento. El caso de uso identificado se coloca en ese o en esos grupos funcionales.

Por ejemplo, en la figura 10, la tarea analizada es inscribir

jugadores, esta tarea según el paso 2, es considerada un caso de uso, por lo tanto para identificar el grupo funcional al que pertenece es necesario analizar el recurso que modifica esta tarea, el cual es: Jugador.

Este grupo funcional fue creado cuando se analizó la dependencia de recurso datos del jugador, entre el SSA y el Jugador. El caso de uso inscribir jugadores se encontrará en el grupo funcional: gestión de jugadores.

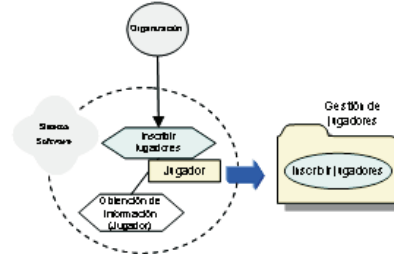


Figura 10. Determinando el grupo funcional de un caso de uso a partir de una relación de interacción

Paso 4. Si la tarea interna del actor sistema de software analizada no se considera un caso de uso, esta tarea puede formar parte de un caso de uso, es decir, puede ser un considerado como un paso, dentro de los escenarios de un caso de uso. Para identificar el caso de uso al que se encontraría ligado, es necesario buscar la tarea más próxima que haya sido considerada como un caso de uso siguiendo un recorrido infijo.

Guía 2.1 Descubriendo los actores de los casos de uso encontrados en el actor sistema de software. Para determinar a los actores del caso de uso encontrados en el actor sistema de software se sigue el mismo procedimiento utilizado para descubrir los actores de los actores de los grupos funcionales (guía 1.2)

Siguiendo con el caso de estudio, la figura 11 muestra el diagrama de casos de uso que resulta del grupo funcional Gestión de jugadores.

Guía 3. Construyendo los escenarios de los casos de uso. El siguiente paso consiste en utilizar los pasos encontrados en las tareas y metas del actor sistema de software (paso 4) para construir los escenarios de cada caso de uso.

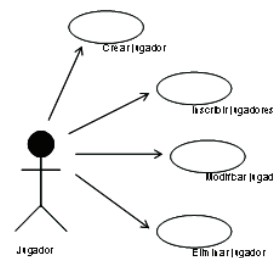


Figura 11. Diagrama de casos de uso: Gestión de Jugadores

6. CONCLUSIONES

Unos de los problemas fundamentales de los trabajos actuales en modelado organizacional es la escasa importancia que se ha dado, en general, al proceso de mapeo entre los elementos del modelo organizacional y los elementos del sistema de software propuesto. La falta de un proceso metodológico para generar sistemas de información a partir de las primitivas de modelado organizacional ocasiona que los esfuerzos que se dedican en modelado el negocio, después no se vean recompensados al generar el sistema de información que dé soporte a la empresa.

Una de las propuestas presentadas en este trabajo es un método para reducir el nivel de abstracción del modelo organizacional, permitiendo de esta forma producir las especificaciones del sistema de información. Este proceso se logra insertando al SSA como un actor en el modelo organizacional y delegando hacia este nuevo actor responsabilidades de los actores de la organización. Por lo tanto el nuevo modelo organizacional generado es un modelo intermedio entre el modelo organizacional y el modelo del sistema de software.

Con la finalidad de reducir la complejidad del proceso de inserción del actor sistema de software, en este trabajo se aborda el uso de patrones de diseño. De esta forma, se propone un patrón para cada una de las distintas posibilidades que se pueden presentar al generar un nuevo modelo organizacional que incluya al actor sistema.

Otra de las propuestas presentadas, es un método que permita derivar las especificaciones de un sistema de software a través de un modelo de casos de uso a partir de un modelo organizacional que incluya al actor sistema de software.

7. RECONOCIMIENTOS

Nos es grato dar reconocimiento a la empresa CARE Technologies y SUPERA (México) por el soporte financiero que ha proporcionado para la realización de este trabajo de investigación.

8. REFERENCIAS

[1] Beedle Michael A. COOherentBPR A pattern language to built agile organizations, Plop-97 Conference, (1997).
[2] Brooks Frederick P., No silver Bullet. Essence and Accidents in Software Engineering. IEEE Computer (1987).
[3] Bubenko, J. A., Jr and M. Kirikova, Worlds in Requirements Acquisition and Modeling, in: Information Modeling and Knowledge Bases VI. H. Kangassalo et al. (Eds.), IOS Press (Amsterdam, 1995), pp. 159-174.
[4] Buschmann, R. Meunier, H. Rohnert, P. Sommerland and M. Stal, Pattern - Oriented software Architecture: A system of Patterns. John Wiley & Sons (1998).
[5] Castro J. Kolp M. Mylopoulos J. Towards Requirements-Driven Information Systems Engineering: The Tropos Project. Information System Journal, Elsevier, Vol 27, (2002), pp 365-389.
[6] Cesare S. Mark Lycett, Business Modelling with UML, distilling directions for future research, Proceedings of the Information Systems Analysis and Specification (Cd. Real,

2002), pp. 570-579.

[7] Cockburn Alistair, Writing Effective Use Cases, Addison-Wesley, ISBN 0-201-70225-8, (2001).
[8] Gamma] E., R. Helm, R. Johnson, and J. Vlissides. Design Patterns, Addison-Wesley, (1995).
[9] Jacobson Ivar, Maria Ericsson, Agneta Jacobson, The Object Advantage: Business process reengineering with object technology, Ed. Addison Wesley, ISBN 0-201-42289-1, (USA, 1995).
[10] Kolp Manuel, Paolo Giorgini, John Mylopoulos: Organizational Patterns for Early Requirements Analysis. Proceedings of the CAiSE 03, (Austria, 2003), pp 617-632.
[11] Kotonya Gerald and Ian Somerville, Requirements Engineering process and techniques, JhonWiley & Sons, USA 1998.
[12] Kulak Daryl Eamonn Guiney, Use Cases requirements in context, Addison-Wesley, ISBN 0-201-65767-8, (2000).
[13] Lamsweerde van A., Requirements engineering in the year 00: A research perspective. In proceeding of the 22nd. IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, (2000). pp. 194-203.
[14] Leite, J.C.S.P. Rossi, G. Balaguer F. Maiorana V., Enhancing a requirements baseline with scenario, Proceedings of the Third IEEE International Symposium on Requirements Engineering RE97, IEEE Computer Society Press, (1997), pp. 44-53.
[15] Martin Robert, Dirk Riehle, Frank Buschmann, and John Vlissides (eds). Patterns Languages of Program Design 3, Addison-Wesley, (1998).
[16] Meszaros G. and J. Doble, A Pattern Language for Pattern Writing, pages 529-574. Pattern Languages of Program Design 3, Addison Wesley, Robert Martin, D. Riehle and F. Buschmann ISBN 0-201-310112, (USA, 1998).
[17] 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, Information Systems 26, (2001).
[18] Piattini G. Mario, Félix O. García, Calidad en el desarrollo y Mantenimiento de Software, Ed. Ra-Ma. (España, 2002).
[19] Rolland R., Souveyet, C., Plihon, V., Method Enhancement with Scenario Based Techniques, Proceedings CAISE 99, 11th Conference on Advanced Information System (1999).
[20] Yu Eric, Modelling Strategic Relationships for Process Reengineering, PhD Thesis, University of Toronto, (Toronto, 1995).

