

Arquitectura basada en modelos para la generación de especificaciones textuales de requisitos a partir de procesos de negocio definidos mediante BPMN

José Manuel Cruz, Begoña Moros Valle, and Ambrosio Toval

Grupo de Investigación de Ingeniería del Software.
Departamento de Informática y Sistemas. Universidad de Murcia.
Campus de Espinardo. 30071. Murcia, España.
{josemanuel.cruz,bmoros,atoval}@um.es

Resumen En la ingeniería del software, la gestión de procesos de negocio (BPM) se suele aplicar para modelar y optimizar los procesos de negocio de un sistema. Junto con la Ingeniería de Requisitos (IR), constituye una base sobre la que especificar un sistema software. Sin embargo, a menudo existe una falta de alineación entre ambas especificaciones que repercute negativamente en el sistema. Este artículo presenta una arquitectura basada en modelos para la generación de especificaciones textuales de requisitos a partir de procesos de negocio representados mediante BPMN y su aplicación a un caso de estudio. El objetivo es agilizar y mejorar la etapa de análisis de un sistema software, generando un subconjunto de los requisitos del sistema para facilitar la escritura de una especificación completa y sincronizada con los procesos de negocio.

Keywords: transformación BPMN-requisitos, generación de requisitos textuales, generación de requisitos basada en modelos, generación de requisitos a partir de procesos de negocio

1. Introducción

En el contexto de la ingeniería del software, la gestión de procesos de negocio (BPM) surge como una etapa inicial del análisis destinada a diseñar, modelar, organizar, documentar y optimizar los procesos de negocio de un sistema. Esta disciplina, que combina las capacidades del software y la experiencia de negocio, constituye una metodología más efectiva, eficiente y tolerante al cambio que un enfoque tradicional centrado en funciones [15], siendo especialmente recomendable para el éxito de sistemas software de gran complejidad. En este contexto, BPMN [19] surge como una notación gráfica, actualmente mantenida por el OMG, destinada a proporcionar un lenguaje común que facilite la comunicación y mejor comprensión de los procesos de negocio. BPMN se ha convertido en un estándar por su sencillez y legibilidad, permitiendo la comprensión de los

procesos de negocio por parte de todos los interesados, tanto dentro como fuera del ámbito de las Tecnologías de la Información (TI). En los últimos años han surgido propuestas para la ejecución directa de procesos de negocio [17], y existen numerosas herramientas que implementan este tipo de funcionalidad [3,5]. Además, la especificación de BPMN 2.0 incluye ya una semántica de ejecución que permite definir procesos de negocio utilizando un formato XML que puede ser ejecutado por herramientas como Activiti [1] y jBPM [4], y que representa la tendencia actual en este campo. No obstante, lo habitual es que al modelado de procesos de negocio le suceda, dentro de la etapa de análisis del sistema, una fase de extracción de requisitos.

La Ingeniería de Requisitos (IR) es una disciplina cuyo principal fin es la gestión de los requisitos de un sistema software, a través de la definición de una especificación (SRS, Software Requirements Specification) en la que se representen las necesidades del cliente. La IR es la base de cualquier desarrollo de software, y para la representación de los requisitos han surgido diferentes notaciones, como los casos de uso o, más recientemente, las historias de usuario presentes en los métodos ágiles. No obstante, la tradicional especificación textual de requisitos, tal y como se define en el estándar IEEE 830 [13] (actualizado al estándar IEEE 29148 [14]) continúa siendo la más utilizada en la industria del software [9].

La falta de alineación entre los modelos de empresa y los de sistemas de información es un problema frecuente, a menudo con consecuencias negativas en el rendimiento de un sistema [12]. De ello se deduce la necesidad de contar con mecanismos que nos permitan mantener la consistencia entre ambas especificaciones. En este sentido, parece natural utilizar las técnicas del Desarrollo de Software Dirigido por Modelos (DSDM) para generar una versión inicial de la especificación de requisitos a partir de los procesos de negocio. En este documento se presenta una arquitectura basada en modelos para la generación de una especificación textual de requisitos a partir de procesos de negocio definidos mediante BPMN. Una arquitectura de este tipo permite agilizar y mejorar la etapa de análisis de un sistema software, facilitando la escritura de un catálogo de requisitos completo y sincronizado con los procesos de negocio. El enfoque propuesto busca automatizar parte del proceso de extracción de requisitos estableciendo la correspondencia entre ambas notaciones. El resultado es un conjunto de requisitos textuales que supondrá un subconjunto del total de requisitos del sistema. El objetivo no es por tanto automatizar todo el proceso, pues esto sería arduo, sino obtener una especificación inicial, consistente con el modelo de negocio, que sirva de guía durante el resto del proceso. Para ello, es recomendable partir de diagramas BPMN que describan procesos de negocio en los que las tareas representan requisitos del sistema y se encuentran definidos con un lenguaje natural. En esta primera versión hemos utilizado una transformación modelo a texto (m2t) para generar la especificación textual de requisitos. Como extensión a esta contribución, un objetivo posterior será el de generar modelos de requisitos conformes al metamodelo de requisitos REMM [16], componiendo así un proceso completo de DSDM.

El resto del artículo está estructurado de la siguiente forma. La Sección 2 realiza un estudio del estado del arte en la transformación de procesos de negocio hacia requisitos. La Sección 3 proporciona una visión general de la solución que se propone para abordar esta transformación. En la Sección 4 se estudian en detalle los aspectos relacionados con la implementación de la arquitectura. La Sección 5 muestra el funcionamiento de la herramienta mediante su aplicación en un caso de estudio real, y la Sección 6 finaliza con las principales conclusiones y un análisis de las vías de futuro del proyecto.

2. Estado del arte

La transformación desde procesos de negocio hacia requisitos ha sido objeto de diversos estudios durante los últimos años. No obstante, como veremos a continuación, el enfoque que se ha hecho en la mayoría de los casos difiere del que aquí presentamos.

En [22] y [21] se propone una transformación desde procesos de negocio representados mediante diagramas UML de actividades hacia requisitos definidos mediante diagramas UML de casos de uso. En estos artículos, los autores proponen un proceso de transformación basado en la identificación de patrones dentro de los diagramas de actividades que tienen su correspondencia en los diagramas de casos de uso. La transformación propuesta en este documento trabaja de una forma similar, a través de la definición de patrones de transformación que, en nuestro caso, definen las equivalencias entre determinadas estructuras del diagrama BPMN y los requisitos textuales generados.

En [11] se propone otra transformación desde diagramas de actividades hacia casos de uso, estableciendo una correspondencia entre los conceptos principales de ambas notaciones e introduciendo el concepto de *step*. Un *step* se define como un conjunto de tareas que un actor concreto puede realizar sin interrupción, y por cada *step* identificado en el diagrama de actividades se genera un caso de uso.

En [6] se trata la transformación desde BPMN hacia casos de uso, de nuevo mediante la identificación de patrones. En este caso no se propone una herramienta que automatice la transformación sino que simplemente se definen una serie de patrones para guiar al experto en el proceso de extracción de casos de uso a partir de procesos de negocio.

También en [10] se estudia este tipo de transformaciones, en este caso proponiendo la generación de requisitos textuales a partir de diagramas EPC extendidos o eEPC (extended Event-driven Process Chain), mientras que en [18] se propone una transformación desde diagramas de características hacia casos de uso.

En general, son propuestas que, como la presentada en este documento, parten de la identificación de patrones y la aplicación de una serie de correspondencias (*mappings*). No obstante, el conjunto de patrones suele ser reducido, en parte por el uso de notaciones poco expresivas (desde el punto de vista de los procesos de negocio) como los diagramas UML de actividades. BPMN define un

conjunto más variado de elementos que nos permite identificar más patrones y extraer más información de un proceso de negocio, trabajando, por ejemplo, con los conceptos de evento y artefacto.

En definitiva, no son muchas las referencias que tratan la transformación desde procesos de negocio hacia requisitos, y existe una carencia particular de estudios que trabajen tanto con especificaciones textuales de requisitos como con BPMN como notación gráfica de procesos de negocio. En el empleo de estas dos notaciones, cuyo uso se encuentra muy extendido en el ámbito empresarial, es donde radica el interés de nuestra propuesta.

Por último, queremos indicar que, aunque no se ha encontrado ninguna herramienta que automatice la transformación desde BPMN hacia especificaciones textuales de requisitos, este tipo de transformación no ha sido completamente ignorada. En [23] se propone una metodología manual para guiar a los expertos durante el proceso completo de extracción de requisitos partiendo de procesos de negocio ya definidos. En [8] se expone un caso de estudio en el que se compararon dos especificaciones de requisitos: una obtenida mediante una técnica convencional de IR y otra derivada (de forma manual) a partir de la definición de un modelo de procesos de negocio del sistema. Se trata en definitiva de estudios que tratan, esta vez sí, la transformación concreta desde BPMN hacia requisitos textuales, pero que lo hacen desde un punto de vista más teórico o manual. La arquitectura presentada en este documento es por tanto, hasta donde sabemos, la primera propuesta para la automatización de este tipo concreto de transformación.

3. Visión general de la solución

La solución propuesta debe ser capaz de obtener especificaciones de requisitos en lenguaje natural a partir de modelos BPMN. Para ilustrar este proceso de transformación, lo más adecuado es empezar por el estudio de ambas notaciones.

El artefacto de entrada consiste en un proceso de negocio representado por un modelo BPMN. Esta notación trabaja principalmente con los conceptos de tarea, puerta (o *gateway* en su término inglés) y evento. Un modelo BPMN representa el flujo de tareas que debe llevarse a cabo y su estructura viene determinada por los arcos que unen las tareas entre sí así como a las puertas y eventos. En un modelo BPMN, la acción a realizar nos la determina el texto de una tarea, mientras que el resto de factores se deducen de la estructura del diagrama. Así, por ejemplo, el encargado de la realización de una tarea se deduce del carril (*lane*) en el que ésta se encuentra, mientras que para averiguar cuándo o bajo qué condiciones se realiza tendremos que estudiar los eventos y los arcos asociados a ella en el diagrama.

Por otro lado, nuestro artefacto de salida será una especificación textual de requisitos. Este documento se confecciona habitualmente siguiendo un determinado estándar, siendo actualmente el más utilizado el IEEE 830 [13] (o su actualización, IEEE 29148 [14]). El objetivo de nuestra solución no es generar una especificación completa sino un listado de requisitos, destinado a ser un

subconjunto del total de requisitos funcionales del sistema. En la representación textual de cada requisito resumiremos en qué consiste la acción, quién y cómo la debe realizar y cuándo o bajo qué condiciones. Además, es posible la definición de una jerarquía de requisitos, con requisitos padre y requisitos hijo que los especializan.

En general, encontramos similitudes entre ambas notaciones, pues ambas representan una información parecida solo que orientada a una etapa diferente dentro del proceso de ingeniería del software, involucrando normalmente a stakeholders distintos. Es bueno que esto sea así pues es la razón de que la transformación tenga sentido y resulte útil, pero a la hora de plantear la automatización de dicha transformación nos encontramos con que el salto semántico entre ambas notaciones es demasiado grande. La implementación de una transformación modelo a texto que genere una SRS directamente a partir del modelo BPMN sería demasiado compleja. La equivalencia entre una tarea en BPMN y un requisito no es siempre directa, y es necesario un cierto recorrido del diagrama para entender las relaciones de las tareas con el resto de elementos e identificar así las diferentes correspondencias que definiremos más adelante en la Sección 4.1. Por tanto, es conveniente estructurar la información necesaria para la transformación a requisitos, de manera que el acceso a los elementos relevantes relacionados con la tarea sea directo.

Por esta razón, se propone la introducción de un metamodelo intermedio, al que llamamos Task Business Process (TBP), cuyo objetivo es servir de puente entre ambas notaciones reduciendo así el salto semántico. La información contenida en este metamodelo puede verse como un subconjunto del metamodelo BPMN pero estructurada de manera que se simplifica la definición de las reglas de transformación, al reducir la complejidad del metamodelo origen. La arquitectura resultante es una transformación realizada en dos pasos, como vemos en la Figura 1, donde son necesarias dos transformaciones: una modelo a modelo (m2m) y otra modelo a texto (m2t). El metamodelo intermedio supone una representación a medio camino entre el modelo BPMN y la especificación textual de requisitos, de forma que se requiere un paso adicional para obtenerlo a partir del modelo BPMN, pero la transformación m2t final se simplifica considerablemente. La introducción de este metamodelo nos permite modularizar el proceso global de transformación, haciéndolo más comprensible y mantenible. Para la implementación de las transformaciones m2m y m2t que componen la arquitectura, hemos utilizado respectivamente los lenguajes ETL y EGL que ofrece el framework Epsilon [2].

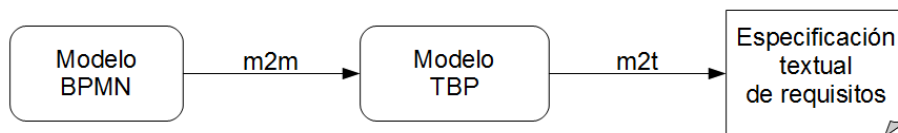


Figura 1. Arquitectura de la solución.

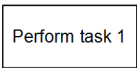

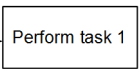
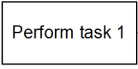

Por último, es necesario explicar cuál va a ser el punto de partida de nuestra solución. BPMN es eminentemente una notación gráfica para el modelado de procesos de negocio, y como tal especifica los diferentes elementos visuales a utilizar y su significado. Pero a la hora de definir diagramas BPMN existen numerosas herramientas y también diferentes formas de serialización. En nuestro caso, proponemos una solución basada en modelos y, como tal, debemos contar con un modelo como artefacto de entrada. Afortunadamente, la especificación oficial de BPMN viene acompañada de un metamodelo que representa todos sus conceptos, destinado al soporte de herramientas como la nuestra. Además, Eclipse cuenta con un editor gráfico para el diseño de diagramas BPMN tal que las estructuras definidas de este modo generan un modelo conforme al metamodelo oficial de BPMN. Estos modelos, serializados en xmi, serán la entrada de nuestra arquitectura.

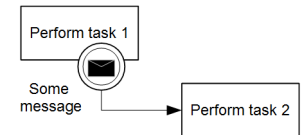
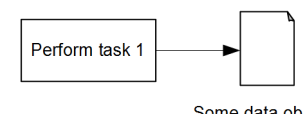
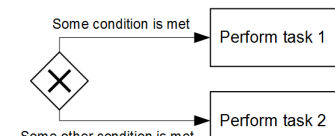
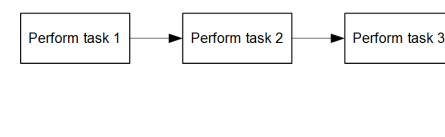
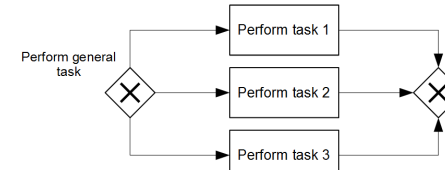
4. Arquitectura propuesta

En este apartado se explicará con más detalle la estructura de la arquitectura propuesta, definiendo las correspondencias entre el artefacto de entrada y salida y el metamodelo intermedio creado para la transformación.

4.1. Definición de las correspondencias (mappings)

El primer paso a tener en cuenta durante el planteamiento de la transformación es la definición de las correspondencias entre elementos del modelo de origen y elementos del artefacto de destino. Se trata de una fase previa a la implementación, donde se deben estudiar ambas notaciones y extraer las equivalencias. En la Tabla 1 pueden verse algunas de las correspondencias definidas para la arquitectura actual. La herramienta generará la especificación de requisitos en inglés, por ser el idioma que favorece una mayor difusión.

nº From*	To
1 	R1. Actor A must perform task 1.
2  → 	R1. Actor A must perform task 1 when the message «some message» is received.
3  → 	R1. Actor A must perform task 1. As a result, the message «some message» must be sent.

<p>4</p> 	<p>R1. Actor A must perform task 1. R1.1. If the message «some message» is received while performing task 1, [this task must be cancelled and]** actor A must perform task 2.</p>
<p>5</p> 	<p>R1. Actor A must perform task 1 making use of some data object.</p>
<p>6</p> 	<p>R1. Actor A must perform task 1 if some condition is met. R2. Actor A must perform task 2 if some other condition is met.</p>
<p>7</p> 	<p>R1. Actor A must perform the following tasks. R1.1. Actor A must perform task 1. R1.2. Actor A must perform task 2. R1.3. Actor A must perform task 3.</p>
<p>8</p> 	<p>R1. Actor A must perform general task. To do so, one of the following alternatives must be chosen: R1.1. Actor A must perform task 1. R1.2. Actor A must perform task 2. R1.3. Actor A must perform task 3.</p>

*Todos los patrones se encuentran dentro de un carril (*lane*) etiquetado como 'Actor A'

**Este fragmento no se generará si el evento intermedio no es de tipo *interrupting*.

Tabla 1: Extracto de las correspondencias definidas en la arquitectura.

La primera de todas ellas es la correspondencia esencial de la arquitectura: la que establece la equivalencia entre una tarea cualquiera y su requisito. Para toda tarea del modelo BPMN se generará un requisito básico donde se especifica el actor que debe realizar la tarea y el contenido de la misma. A continuación, se definen una serie de correspondencias que extienden a la primera identificando en el diagrama BPMN patrones un poco más complejos que el que representa una tarea aislada. Se trata, por ejemplo, de patrones en los que se tienen en cuenta los eventos que pueden ir asociados a una tarea. Así, contamos con correspondencias que nos indican el requisito textual que debe generarse cuando tenemos una tarea precedida de un evento (correspondencia n°2), cuando el evento es posterior a la tarea (correspondencia n°3) y cuando se trata de un evento intermedio (correspondencia n°4). El texto generado será diferente según el tipo de evento asociado.

De la misma forma, identificamos en el modelo BPMN de entrada la posible asociación entre una tarea y un artefacto (correspondencia n°5), relación que representaremos también en la especificación textual del requisito generado. Otra correspondencia importante es la que nos determina el texto a generar a partir de una tarea precedida por una puerta exclusiva (correspondencia n°6). Lo habitual es que al utilizar este tipo de puertas en un diagrama BPMN se etiqueten los diferentes arcos para especificar bajo qué condiciones se debe tomar una u otra rama. Identificar este patrón es importante para que en la especificación textual del requisito se incluyan este tipo de precondiciones.

Las dos últimas correspondencias definidas son un poco más complejas, pues el objetivo no es generar un único requisito sino una jerarquía. La primera de ellas (correspondencia n°7) establece que, ante una secuencia de tareas en el diagrama BPMN, debemos generar un requisito para cada una de ellas y otro requisito padre que las englobe. El objetivo es introducir requisitos genéricos que nos permitan representar con más claridad diagramas en los que exista una gran secuenciación en las tareas. La definición de requisitos padre permitirá al usuario, si así lo desea, eliminar los requisitos hijo cuando estos sean demasiado específicos, consiguiendo una granularidad mayor en la especificación.

La última correspondencia (correspondencia n°8) también propone la obtención de una jerarquía de requisitos, sólo que en esta ocasión tiene un significado mucho más concreto. Se trata de un patrón compuesto por una puerta exclusiva que define dos o más ramas formadas por tareas, las cuales convergen en una nueva puerta exclusiva. Es una estructura utilizada habitualmente en la definición de diagramas BPMN cuando se quiere expresar una acción que puede ser realizada de diferentes maneras, donde cada una de las ramas representa una alternativa para realizar dicha acción. Se puede apreciar por tanto la utilidad de representar explícitamente esta situación en la especificación de requisitos. Para ello, lo que haremos será generar un requisito padre que representa la acción global, y cuyo enunciado aparecerá normalmente como una etiqueta en la puerta exclusiva. Después, los requisitos generados para cada una de las ramas se establecerán como requisitos hijo. El texto del requisito padre dejará clara la condición de alternativa que tiene cada requisito hijo.

4.2. Metamodelo Task Business Process (TBP)

En la Figura 1 exponíamos el esquema de transformación en dos pasos de nuestra arquitectura, caracterizado por la introducción de un metamodelo intermedio. Este metamodelo, al que llamamos Task Business Process (TBP) y que podemos ver en la Figura 2, tiene como principal objetivo reducir el salto existente entre los modelos BPMN y los requisitos textuales. Por tanto, el metamodelo TBP consistirá en una representación intermedia entre ambas notaciones. El objetivo es que la especificación textual de requisitos pueda obtenerse de manera sencilla a partir del modelo TBP. Para ello, los conceptos que aparecen en BPMN y que son relevantes para un requisito aparecerán representados en el metamodelo y estructurados de la forma más adecuada. Durante este proceso prescindiremos de muchos de los elementos definidos en el metamodelo BPMN

que resultan innecesarios por ser irrelevantes para la generación de requisitos (conceptos relacionados con la ejecución de los modelos tales como coreografía, orquestación o llamada) o por no estar soportados por la herramienta (como la definición de parámetros y propiedades o la extensa jerarquía de tareas que define BPMN), obteniendo así un metamodelo más sencillo y mejor orientado a la generación de requisitos.

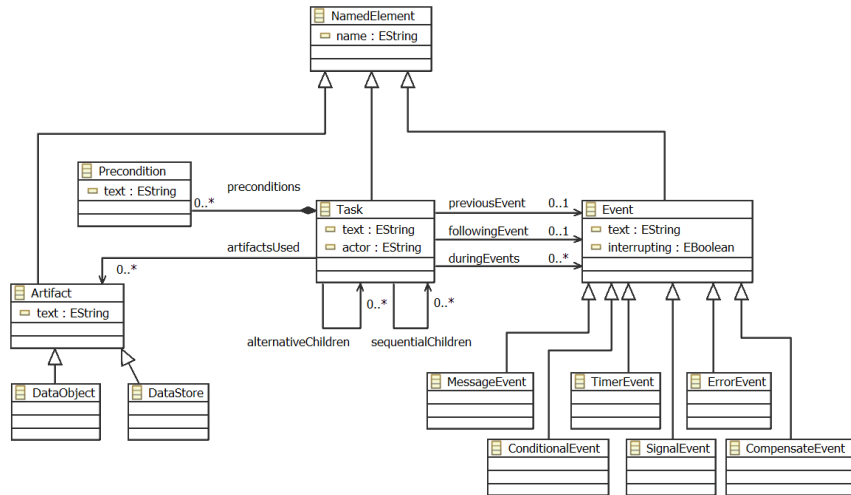


Figura 2. Metamodelo TBP.

El metamodelo TBP tiene como concepto central el de tarea (Task). Todo diagrama BPMN será traducido en un conjunto de tareas, cada una de las cuales acabará generando un requisito textual. Las tareas definidas en el metamodelo TBP tienen dos atributos: el texto y el actor que las realiza, y se permite la jerarquía de tareas a través de las relaciones *sequentialChildren* y *alternativeChildren*. Esto nos permite representar tareas formadas por la secuencia o alternativa de otras tareas, y está relacionado con los dos últimos patrones estudiados en la Sección 4.1. Además, se definen los conceptos de artefacto y evento, contemplados en el metamodelo BPMN, y el concepto de precondition, todos ellos asociados a una tarea. La precondition es un concepto de gran utilidad, que no existe como tal en BPMN pero que introducimos en nuestra arquitectura para representar este tipo de condiciones, las cuales suelen aparecer en forma de arcos etiquetados que unen una tarea con una puerta exclusiva, situación que vemos representada en la correspondencia n°6 de la Tabla 1. Por otro lado, para artefactos y eventos definimos una jerarquía de subclases representando la mayoría de tipos permitidos en BPMN. Esta jerarquía es especialmente numerosa en el caso de los eventos (a pesar de que el metamodelo no contempla la totalidad de eventos existentes en BPMN), lo cual es una buena muestra de la expresividad de esta notación. Además, representamos para cada evento las diferentes formas en que

se puede relacionar con una tarea, apareciendo antes, durante, o después de la misma.

En definitiva, el metamodelo TBP contempla los conceptos más importantes de un diagrama BPMN reestructurando el proceso de negocio para representarlo como una jerarquía de tareas. Las relaciones entre los distintos conceptos se extraen a partir de un recorrido del diagrama BPMN en el que se analizan los arcos que unen los diferentes nodos. Este recorrido nos permite identificar ciertos conceptos que no estaban presentes en el metamodelo BPMN original y que resultan de gran utilidad en la generación de requisitos, como la jerarquía de tareas (alternativas y secuenciales) o el concepto de precondition. La extracción de esta información entraña cierta complejidad y representa una de las principales ventajas de estructurar la transformación en dos pasos. La introducción del metamodelo intermedio nos permite centrarnos en los aspectos necesarios para la generación de un requisito, y la dificultad de extraer éstos u otros conceptos que pudiéramos añadir en un futuro queda restringida a la transformación modelo a modelo inicial. Una vez obtenida la representación del proceso de negocio conforme al metamodelo TBP, la generación de requisitos se simplifica considerablemente, pues cada tarea tiene toda la información que necesita para generar de forma completa el requisito textual correcto.

5. Validación de la propuesta

Para comprobar el funcionamiento de la arquitectura se ha seguido un sencillo proceso en el que se ha partido de siete diagramas BPMN correspondientes a una especificación real de los procesos de negocio de un sistema software denominado TicTae [20]. TicTae es un proyecto para una red social profesional especializada en el sector TIC, especialmente orientada a la implicación directa de los centros educativos. Se ha creado un proyecto piloto con el Servicio de Informática de la Universidad de Murcia. Los diagramas utilizados tratan concretamente con el proceso de intercambio de información académica entre los centros educativos y el propio sistema. Estos diagramas han sido reproducidos utilizando el editor gráfico para BPMN proporcionado por Eclipse, permitiéndonos obtener directamente los modelos. Durante este proceso se ha modificado ligeramente la forma de nombrar los componentes del diagrama para facilitar que los requisitos se generen con un lenguaje natural. Normalmente, esto no supuso más que alterar un poco la forma en que se enuncian las tareas, pues es habitual, por ejemplo, prescindir de artículos para simplificar el texto. Una vez obtenidos los siete modelos BPMN correspondientes a los diagramas, se ha suministrado cada uno de ellos como entrada a la transformación definida, obteniendo los correspondientes requisitos textuales.

En este apartado analizaremos uno de los diagramas BPMN tomados como ejemplo para la validación, que podemos ver en la Figura 3. Se trata de un proceso de negocio en el que se especifica la manera en que se gestiona un cambio en el sistema TicTae.

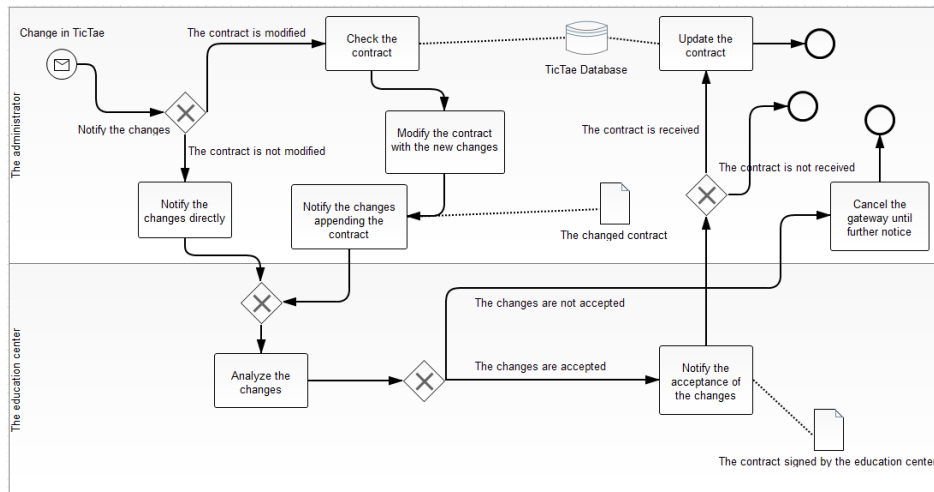


Figura 3. Diagrama BPMN de ejemplo del sistema TicTae.

El diagrama es sencillo pero cuenta con los suficientes recursos expresivos para mostrar las características de la arquitectura. Existen eventos, artefactos, tareas secuenciales y también un patrón alternativo. El resultado de ejecutar la transformación sobre este modelo BPMN es el que vemos a continuación:

Requisitos textuales generados para el diagrama BPMN de ejemplo

- R1.The system must notify the changes in TicTae.
- R1.1.The administrator must cancel the gateway until further notice if the changes are not accepted.
- R1.2.The administrator must notify the changes when the message "change in TicTae" is received.
- To do so, one of the following alternatives must be chosen:
- R1.2.1.The administrator must perform the following tasks.
- R1.2.1.1.The administrator must check the contract if the contract is modified, making use of ticTaeDatabase.
- R1.2.1.2.The administrator must modify the contract with the new changes.
- R1.2.1.3.The administrator must notify the changes appending the contract, making use of the changed contract.
- R1.2.2.The administrator must notify the changes directly if the contract is not modified.
- R1.3.The education center must notify the acceptance of the changes if the changes are accepted, making use of the contract signed by the education center.
- R1.4.The administrator must update the contract if the contract is received, making use of ticTaeDatabase.
- R1.5. The education center must analyze the changes.

Analizando con detenimiento cada uno de los requisitos generados se puede comprobar la forma en que se representan los diferentes conceptos que hemos

ido viendo a lo largo del documento: eventos, precondiciones, uso de artefactos, la jerarquía de tareas, etc. El patrón alternativo, por ejemplo, ha sido correctamente identificado, como vemos en el correspondiente requisito R1.2. Este requisito nos permite además comprobar el texto generado cuando tenemos un evento anterior a una tarea. Como requisitos hijo se definen R1.2.1 y R1.2.2, que son formas alternativas de cumplir el requisito R1.2. El requisito R1.2.1 es, además, un ejemplo de cómo funciona la agrupación secuencial de tareas, pues está compuesto a su vez de otros tres requisitos hijo.

En definitiva, el conjunto final de requisitos obtenido representa una versión inicial de la funcionalidad que ha de tener el sistema. En muchos casos será conveniente una revisión posterior para refinar y completar la especificación de requisitos del sistema a desarrollar, adaptando ésta al estilo habitual de redacción de requisitos del equipo.

6. Conclusiones y vías futuras

Hemos presentado una arquitectura generativa que persigue la correcta alineación entre los modelos de empresa y los de sistemas de información, a través de la generación de requisitos desde procesos de negocio. Este tipo de transformación ha recibido escasa atención, si bien en los últimos años han aparecido algunas publicaciones al respecto en la literatura. Este artículo constituye una propuesta especialmente original por el uso de BPMN como notación para representar los procesos de negocio y la especificación textual para la definición de los requisitos, dos notaciones de uso extendido en el ámbito industrial. En concreto, el uso de BPMN proporciona una riqueza expresiva que nos permite generar requisitos textuales variados y completos, gracias al uso de los recursos definidos en BPMN y a la identificación de patrones. Esta arquitectura nos permite agilizar y mejorar la etapa de análisis de un sistema software, ayudando al ingeniero de requisitos con la generación automática de un subconjunto inicial de los requisitos que, aunque tengan que ser revisados posteriormente, suponen una primera versión completa y sincronizada con los procesos de negocio. Como valor añadido, esta expresión textual del modelo BPMN puede ayudar a entender mejor este tipo de diagramas por parte de stakeholders no expertos en dicha notación. En este sentido, la herramienta sigue la línea marcada por otros trabajos como [7], en los que se define la transformación hacia especificaciones con un carácter menos técnico.

La transformación implementada supone una primera aproximación al problema, y como tal existen diversas modificaciones y extensiones posibles. La arquitectura actual considera los elementos principales de un diagrama BPMN, pero no tiene en cuenta todos ellos, de manera que determinados conceptos como los subprocesos o las transacciones no están soportados. Además, no se distingue entre los diferentes tipos de artefactos y tareas existentes, algo que podría ser útil en diagramas BPMN más complejos donde no todas las tareas deben traducirse en requisitos.

La agrupación de tareas también puede ser mejorada, realizando un análisis más avanzado del flujo de tareas de un diagrama BPMN para permitir identificar patrones más complejos que los actuales. En este sentido sería también interesante establecer, cuando sea posible, el orden que deben seguir las tareas secuenciales, pues esta información está disponible en el diagrama BPMN pero se pierde durante el proceso actual de transformación. Otras funcionalidades que podrían añadirse a la arquitectura actual incluyen la búsqueda recursiva de precondiciones y eventos o la implementación de una jerarquía de actores.

Por otro lado, sería interesante integrar el sistema con herramientas existentes para el modelado de procesos con BPMN. De esta forma, no sería necesario crear los diagramas con el editor gráfico de Eclipse sino que podríamos inyectar diagramas ya generados con otras herramientas, obteniendo el modelo correspondiente para suministrarlo como entrada de la transformación. Una extensión más sencilla sería la adaptación de la arquitectura para que genere la especificación textual de requisitos en base a algún formato estándar de definición de requisitos, en lugar de como texto plano.

Por último, y como ya comentamos en la introducción, podríamos prescindir de la especificación textual y, en su lugar, contar con un modelo de requisitos como destino de la transformación. Siguiendo esta línea, planeamos integrar nuestra herramienta con el metamodelo de requisitos REMM [16], construyendo un esquema completamente basado en modelos en el que la actual transformación m2t sería sustituida por una segunda transformación m2m. Una vez refinada y completada la herramienta, sería interesante llevar a cabo una validación empírica de la misma a través de un caso de estudio que involucre a usuarios reales.

7. Agradecimientos

Esta investigación ha sido parcialmente financiada por el proyecto GEODAS-REQ (TIN2012-37493-C03-02) del Ministerio de Economía y Competitividad, y fondos FEDER europeos.

Referencias

1. Activiti bpm platform. <http://activiti.org/>
2. Epsilon. <http://eclipse.org/epsilon/>
3. Intalio bpms. <http://www.intalio.com/products/bpms/overview/>
4. jbpm. <http://www.jbpm.org/>
5. Oracle business process management suite 12c. <http://www.oracle.com/us/technologies/bpm/suite/overview/index.html>
6. Berrocal, J., Alonso, J.M.G., Rodríguez, J.M.M.: Patrones para la extracción de casos de uso a partir de procesos de negocio. In: II Taller de Procesos de Negocio e Ingeniería de Servicios. pp. 1–11 (2009)
7. Cabot, J., Pau, R., Raventós, R.: From uml/ocl to sbvr specifications: A challenging transformation. *Information Systems* 35(4), 417 – 440 (2010),

- <http://www.sciencedirect.com/science/article/pii/S030643790800094X>, vocabularies, Ontologies and Rules for Enterprise and Business Process Modeling and Management
8. Cardoso, E., Almeida, J., Guizzardi, G.: Requirements engineering based on business process models: A case study. In: Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th. pp. 320–327 (Sept 2009)
 9. Chernak, Y.: Requirements reuse: The state of the practice. In: Software Science, Technology and Engineering (SWSTE), 2012 IEEE International Conference on. pp. 46–53 (June 2012)
 10. Coskunçay, A.: An approach for generating natural language specifications by utilizing business process models. Master Thesis (2010), <http://etd.lib.metu.edu.tr/upload/12612305/index.pdf>
 11. Dijkman, R.M., Joosten, S.M.M.: Deriving use case diagrams from business process models. Technical Report TR-CTIT-02-08, Centre for Telematics and Information Technology University of Twente, Enschede (March 2002)
 12. Heinrich, R., Merkle, P., Henss, J., Paech, B.: Integrating business process simulation and information system simulation for performance prediction. Software and Systems Modeling pp. 1–21 (2015), <http://dx.doi.org/10.1007/s10270-015-0457-1>
 13. IEEE: ISO/IEC/IEEE 830: IEEE Recommended Practice for Software Requirements Specifications (1998)
 14. IEEE: ISO/IEC/IEEE 29148: Systems and software engineering – Life cycle processes –Requirements engineering (2011)
 15. Ko, R.K.L.: A computer scientist’s introductory guide to business process management (bpm). Crossroads 15(4), 4:11–4:18 (Jun 2009), <http://doi.acm.org/10.1145/1558897.1558901>
 16. Moros, B., Toval, A., Rosique, F., Sánchez, P.: Transforming and tracing reused requirements models to home automation models. Information and Software Technology 55(6), 941 – 965 (2013), <http://www.sciencedirect.com/science/article/pii/S0950584912002388>
 17. OASIS: Web Services Business Process Execution Language (WS-BPEL), Version 2.0 (2007), <http://docs.oasis-open.org/wsbpel/2.0/0S/wsbpel-v2.0-0S.html>
 18. de Oliveira, R.P., Blanes, D., Gonzalez-Huerta, J., Insfran, E., Abrahão, S., Cohen, S., de Almeida, E.S.: Defining and validating a feature-driven requirements engineering approach. Journal of Universal Computer Science 20(5), 666–691 (may 2014), http://www.jucs.org/jucs_20_5/defining_and_validating_a
 19. OMG: Business Process Model and Notation (BPMN), Version 2.0 (2011), <http://www.omg.org/spec/BPMN/2.0/>
 20. Sola, J.M.: Tic talent europe - análisis y desarrollo de infraestructura tecnológica para una nueva red social profesional. Trabajo Fin de Grado (2014)
 21. Štolfa, S., Vondrák, I.: A description of business process modeling as a tool for definition of requirements specification. In: Proceedings of System Integration 2004, Ed. Jan Pour. pp. 463–469 (2004)
 22. Štolfa, S., Vondrák, I.: Mapping from business processes to requirements specification. Tech. rep. (2013), http://www.researchgate.net/publication/242409175_Mapping_from_Business_Processes_to_Requirements_Specification
 23. de la Vara, J., Sánchez, J.: Improving requirements analysis through business process modelling: A participative approach. In: Abramowicz, W., Fensel, D. (eds.) Business Information Systems, Lecture Notes in Business Information Processing, vol. 7, pp. 165–176. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-79396-0_15