

A Model-Driven Approach for Documenting Business and Requirements Interdependencies for Architectural Decision Making

J. Berrocal, J. Garcia Alonso, C. Vicente Chicote and J. M. Murillo

Abstract— In business and system requirements analysis, the characteristics of the elements identified are perfectly documented with specific notations and in specific artifacts. However, since these notations are focused on each specific type of element, the interdependencies between elements of different kinds are usually left implicit. These relationships are particularly important during system design in order to evaluate the impact of each requirement and to select the architectural pattern that better satisfies them. To identify and make them explicit, the architect has to analyze all the artifacts generated in depth. Any misinterpretation of these relationships may lead to patterns being selected that can hinder rather than facilitate the satisfaction of the business goals and the system requirements. This paper presents a set of profiles allowing designers to explicitly model these interdependencies in BPMN 2 and UML 2 Use Case diagrams. In addition, ATL transformations are defined to automatically derive these relationships from the business specification to the requirements models, facilitating their analysis by the architect, and thus reducing the risk of misinterpretation.

Keywords— Business-IT alignment, Business Process Models, Requirements Models, Requirements Relationships.

I. INTRODUCCIÓN

EL DESARROLLO de un sistema software comienza con el análisis minucioso del negocio de la organización con el fin de identificar los objetivos, estrategias y procesos del negocio [1]. Esta información es básica para posteriormente obtener los requisitos del sistema alineados con el negocio [2].

Metodologías, como las detalladas por Gao [3] o Ullah [4], guían en la identificación y análisis de los distintos tipos de información del negocio y su posterior documentación en artefactos específicos. Así, por ejemplo, los objetivos de negocio son documentados en el *Modelo de Objetivos*, y los procesos de negocio en el *Modelo de Procesos de Negocio* [4]. De esta forma, todas las características de cada tipo de información son encapsuladas en artefactos específicos, siendo más fácil su documentación y subsiguiente localización. No obstante, esta separación dificulta el modelado de las relaciones,

o interdependencias, entre elementos de distinto tipo. Así, nativamente con las notaciones utilizadas para modelar los elementos del negocio no se puede detallar qué objetivos influyen sobre cada parte de los procesos de negocio, información ésta que resulta vital para capturar correctamente las necesidades del negocio.

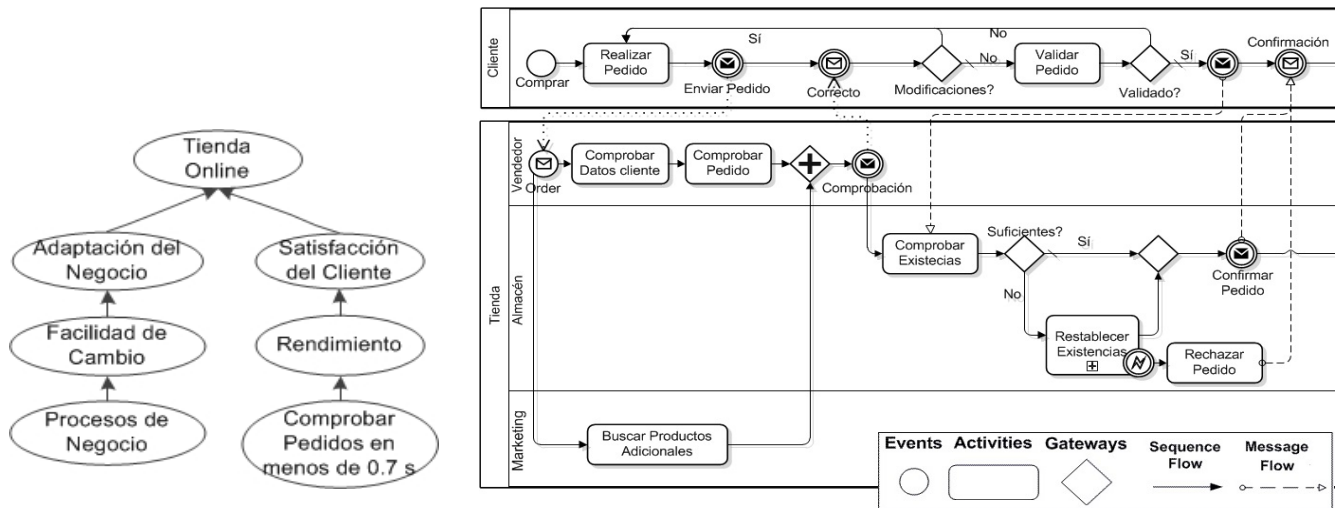
Los artefactos generados son más tarde analizados para definir los requisitos del sistema alineados con el negocio [5]. Los requisitos no funcionales normalmente se identifican mediante el análisis de los objetivos de negocio, y son detallados en el *Documento de Especificación Complementaria*. Por su parte, los requisitos funcionales se obtienen mediante el análisis de los procesos de negocio, para su posterior especificación en el *Modelo de Casos de Uso* [6]. De nuevo, cada tipo de requisito es documentado en artefactos y con modelos específicos, lo que facilita su documentación pero dificulta la especificación de las interdependencias entre ellos.

Durante el diseño de la arquitectura del sistema, es necesario tener en cuenta estas relaciones. El arquitecto necesita conocer qué funcionalidades están restringidas por los distintos atributos de calidad [7]. De esta forma, puede tomar las decisiones arquitectónicas que mejor faciliten el cumplimiento de ambos [8]. Sin embargo, debido a que cada artefacto está centrado en tipos de elementos específicos, las relaciones entre ellos normalmente se encuentran detalladas de forma implícita, esto es, ocultas en los modelos gráficos y descripciones textuales [9], dejando al arquitecto la tarea de descubrirlas.

Para identificarlas, el arquitecto debe consolidar todos los artefactos. Esta consolidación es complicada y costosa [10]. Primero, el arquitecto debe conocer el contenido de cada documento y, después, debe identificar las relaciones entre los elementos. La correcta identificación de estas relaciones depende en gran medida de su experiencia y habilidad. Si no son correctamente identificadas se aumentan las probabilidades de que tome decisiones arquitectónicas que dificulten, o incluso impidan, la alineación del sistema con el negocio y la consecución de los objetivos del mismo.

El trabajo presentado en este artículo facilita el modelado y derivación de estas interdependencias. Para ello, propone (1) un conjunto de anotaciones para BPMN 2 [11] que permiten reflejar las relaciones entre los procesos y los objetivos de negocio, así como información relevante para luego derivar los requisitos y sus interdependencias; (2) transformaciones

J. Berrocal, Universidad de Extremadura, España, jberolm@unex.es
 J. Garcia Alonso, Universidad de Extremadura, España, jgaralo@unex.es
 C. Vicente Chicote, Universidad de Extremadura, España, cristinav@unex.es
 J. M. Murillo, Universidad de Extremadura, España, juanmamu@unex.es



(a) Objetivos de negocio.

(b) Extracto del proceso de negocio para gestionar pedidos.

Figura 1. Modelos de negocio del ejemplo de la tienda online.

ATL [12] que automatizan el proceso de extracción de los requisitos y sus relaciones a partir de los procesos de negocio anotados; y (3) un perfil para el Diagrama de Casos de Uso de UML 2 que permite el modelado de las relaciones entre Casos de Uso y atributos de calidad. De esta forma, se pueden modelar dichas interdependencias tanto a nivel de negocio como de requisitos, además de ser automáticamente derivadas de una fase a otra. Esta información permite reducir los riesgos de malas interpretaciones de las interdependencias y facilita la toma de decisiones arquitectónicas. El trabajo presentado en este artículo se centra en el modelado de estas relaciones y su derivación automática a un modelo de requisitos. En [13] y [14] se detalla cómo se reutilizan estas relaciones para mejorar la toma de decisiones arquitectónicas.

El resto del artículo está organizado de la siguiente manera. La sección II presenta las motivaciones y los trabajos relacionados. La sección III detalla el proceso de modelado y derivación de las relaciones. La sección IV contiene la validación del trabajo en un proyecto real. La sección V presenta las lecciones aprendidas a través de la validación del trabajo y, por último, la sección VI contiene las conclusiones.

II. TRABAJOS RELACIONADOS

El Desarrollo Dirigido por el Negocio (Business Driven Development, BDD) es una disciplina cuyo objetivo es el desarrollo de sistemas IT alineados con el negocio [15]. Propone documentar la información del negocio y reutilizarla como fuente para la generación de los requisitos y el diseño del sistema. Trabajos, como [3] y [16], dan soporte a esta disciplina definiendo notaciones y metodologías que guían durante el análisis del negocio. Usualmente, para modelar los objetivos de negocio utilizan notaciones similares a los árboles de Goals [17], y para documentar los procesos de negocio utilizan BPMN [11], ya que es una notación ampliamente aceptada [18].

Para demostrar la aplicación de estas técnicas se va a seguir

un ejemplo de una tienda online. Esta tienda quiere desarrollar un sistema que de soporte a la recepción y gestión de pedidos a través de Internet. Sus ventajas competitivas serán: primero, la agilidad en la respuesta al cliente en la tramitación de pedidos y, segundo, la facilidad de modificación de los procesos de negocio, y también del sistema, para mejorar la interacción con el cliente o la gestión de pedidos.

Si aplicamos alguna de las técnicas anteriores a este ejemplo, se generarían al menos dos artefactos: uno que documentaría los objetivos y otro los procesos de negocio. La Fig. 1 muestra un extracto de dichos artefactos. Así, en la Fig. 1a se detallan los objetivos que indican, por un lado, que los procesos deben ser fáciles de evolucionar y, por otro, que la comprobación de los pedidos debe ser realizada en menos de 0.7 segundos. La Fig. 1b presenta un extracto del proceso de negocio para gestionar los pedidos.

Si bien en ambos artefactos los distintos elementos quedan perfectamente documentados, no es posible modelar explícitamente las relaciones entre los objetivos y las tareas o procesos de negocio que deben cumplirlos. Así, en el proceso de la tienda online no se puede modelar que todo el proceso debe ser fácil de evolucionar y que exactamente las tareas de *Comprobación de los datos del pedido* y de *Búsqueda de Productos Adicionales* deben ser realizadas en menos de 0.7 segundos para que el cliente pueda realmente percibir la eficiencia. Durante el análisis del negocio, la identificación de estas relaciones puede llegar a ser relativamente simple. Sin embargo, si no son documentadas, su identificación en sucesivas fases del desarrollo se hace más compleja, ya que requiere una mayor experiencia y conocimiento de los artefactos generados.

En [19] y en [20] se definen técnicas para relacionar procesos con objetivos de negocio. Los autores definen metodologías para identificar, descomponer y modelar objetivos del negocio, mediante árboles de Goals, hasta que son operacionales. A partir de los Goals operacionales se identifican nuevas tareas que son incluidas en los procesos. Dichas tareas podrían ser utilizadas para derivar relaciones entre objetivos

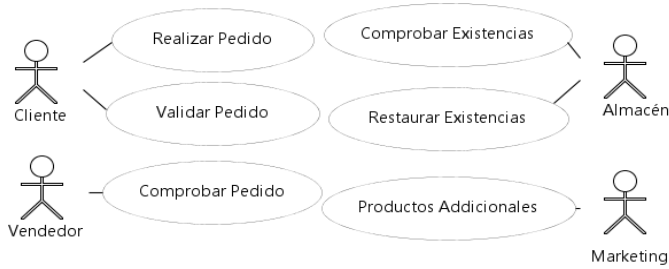


Figura 2. Diagrama de Casos de Uso de la tienda online.

operacionales y procesos de negocio. Sin embargo, no se podrían utilizar para detallar las relaciones con restricciones ni objetivos no operacionales. Así, en el ejemplo de la tienda online, no se podrían detallar las relaciones entre los objetivos definidos y el proceso de gestión de pedidos. Esto se debe a que, por ejemplo, la comprobación de un pedido en un determinado tiempo no implica la realización de nuevas tareas, sino restricciones sobre las ya existentes.

En [21] los autores indican cómo modelar restricciones operacionales en procesos de negocio con el fin de poder definir controles que gestionen dichas restricciones. Para ello, extienden BPMN con dos nuevos elementos: *Operating Condition*, para poder anotar las restricciones operacionales, y *Control Case*, para definir mecanismos de control. En el proceso de la tienda online, se podría añadir un *Operating Condition* indicando el tiempo en el que debe ser comprobado un pedido y un *Control Case* para gestionar esta restricción. De esta forma, dicha relación quedaría documentada. Sin embargo, estos elementos no pueden ser utilizados para detallar restricciones no-operacionales. Así, para la tienda online, la restricción de mantenibilidad no podría ser documentada.

Para dar soporte a una fase sucesiva de la disciplina BDD existen trabajos centrados en la obtención de requisitos a partir del negocio. Trabajos como los detallados en [22], [23], [24] y [25] definen cómo derivar requisitos funcionales a partir de procesos de negocio. De este modo, se mejora la alineación entre los requisitos y el negocio. Por tanto, si aplicamos dichos trabajos al proceso de la tienda online (Fig. 1b) se obtendría un conjunto de Casos de Uso como el mostrado en la Fig. 2. Si bien la obtención de este diagrama proporciona un gran beneficio, estas propuestas no definen cómo identificar las relaciones entre requisitos funcionales y atributos de calidad. Para obtener una alineación más completa, sería deseable poder obtener y modelar estas relaciones.

En [26] los autores utilizan notas en lenguaje natural para documentar las relaciones entre Casos de Uso y atributos de calidad. Así, en el diagrama de la tienda online se podrían añadir estas notas para indicar que todos los Casos de Uso deben ser fáciles de adaptar, y que exactamente los Casos de Uso *Comprobar Pedidos* y *Productos Adicionales* deben cumplir la restricción de eficiencia. Sin embargo, en primer lugar, no se indica cómo obtener y derivar dichas interdependencias a partir del negocio y, en segundo lugar, estas notas, al estar en lenguaje natural, no pueden ser reutilizadas por herramientas que asistan al arquitecto en la toma de decisiones

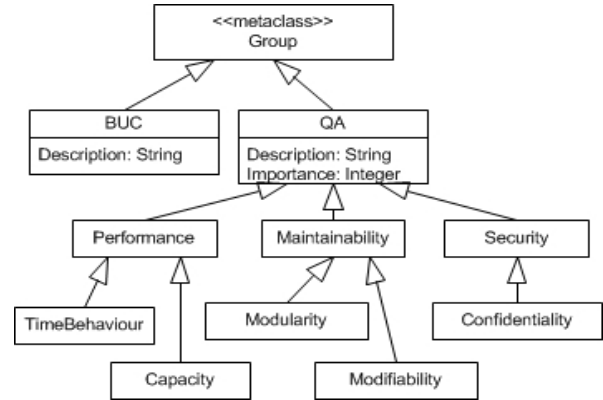


Figura 3. Extracto de la extensión de BPMN.

en las sucesivas fases del desarrollo [27].

Detallar un conjunto de requisitos alineado con el negocio, así como sus relaciones, es muy importante para que el arquitecto pueda hacer un correcto diseño del sistema [7], seleccionando los patrones arquitectónicos que mejor permitan satisfacerlos. De este modo, para la tienda online, conociendo exactamente qué Casos de Uso deben cumplir el requisito de rendimiento, se pueden aplicar patrones específicos para mejorar la eficiencia de sus algoritmos. En este sentido, el trabajo presentado en este artículo se centra en obtener dicho conjunto de requisitos e interdependencias. Además, como se muestra en [13] y [14], dicho conjunto es generado de forma que puede ser reutilizado para asistir al arquitecto en la toma de decisiones arquitectónicas.

III. MODELADO DE LAS RELACIONES ENTRE ELEMENTOS DEL NEGOCIO Y REQUISITOS

Una vez que el negocio ha sido analizado, y se ha generado la documentación del mismo, se deben documentar las relaciones entre objetivos y procesos de negocio. Para ello, se ha creado una extensión de BPMN que permite anotar estas interdependencias directamente sobre los elementos de flujo modelados en cada proceso de negocio.

Estas anotaciones son posteriormente reutilizadas para derivar los modelos de requisitos, junto con las interdependencias entre requisitos funcionales y no-funcionales, o atributos de calidad. Para ello, se han definido un conjunto de transformaciones ATL, que permiten automatizar la derivación de esta información, y un perfil para los diagramas de Casos de Uso de UML, para su modelado y documentación.

Estas extensiones y perfiles se han diseñado para que puedan ser reutilizados por herramientas que asistan al arquitecto en la toma de decisiones arquitectónicas y, así, facilitar el diseño de sistemas alineados con el negocio.

A. Anotando Procesos de Negocio

Una vez que los objetivos y procesos de negocio han sido identificados y documentados mediante el uso de las técnicas indicadas anteriormente, el experto de negocio y el ingeniero de requisitos deben documentar las interdependencias entre objetivos y procesos de negocio. Para ello, deben detallar qué

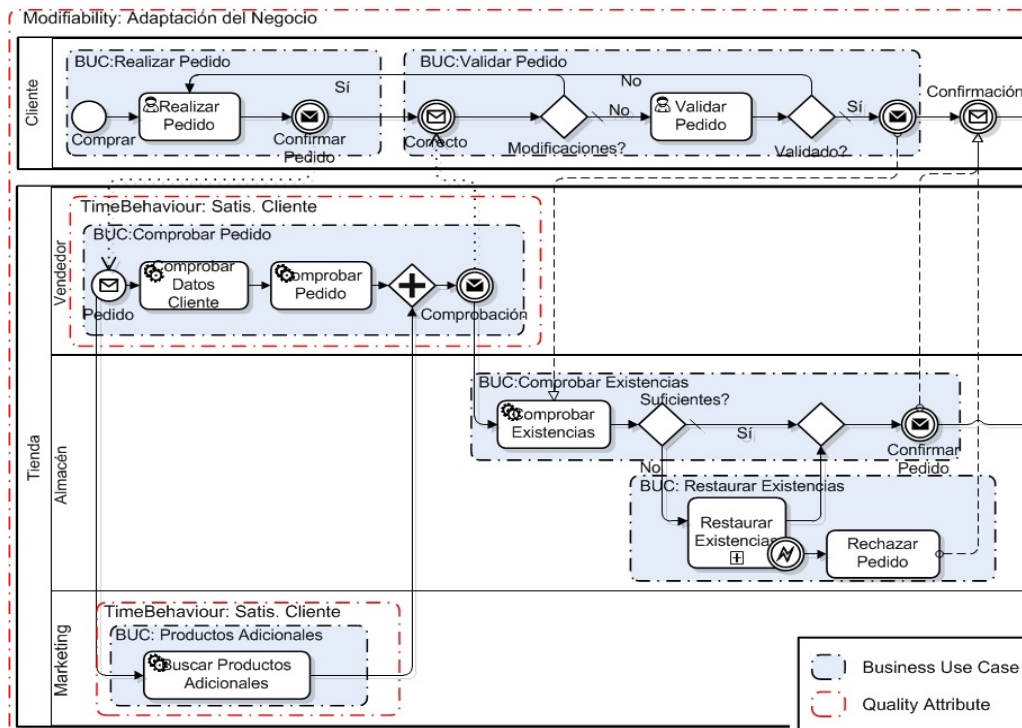


Figura 4. Proceso de negocio para gestionar pedidos anotado con los BUCs y las relaciones con los atributos de calidad.

parte de los procesos de negocio deben cumplir los distintos objetivos o atributos de calidad.

Debido a que esta información no puede ser modelada nativamente en BPMN, se ha extendido el artefacto *Group* para que permita agrupar elementos del flujo del proceso según el atributo de calidad que deben cumplir. La Fig. 3 muestra un extracto de dicha extensión¹, donde se detalla cómo el elemento *Group* es extendido por el elemento *QA* (*Quality Attribute*), el cual es a su vez extendido por un conjunto de sub-atributos siguiendo el modelo de calidad definido por la ISO/IEC 25010 [28]. Esta extensión ha sido aplicada al plugin de Eclipse BPMN2 Modeler [29].

La Fig. 4 muestra el proceso de la tienda online anotado. Como se ha comentado anteriormente, este proceso está relacionado con dos objetivos: debe ser fácil de adaptar a los cambios en el negocio, y la comprobación de los pedidos debe ser realizada en un tiempo determinado para que el cliente esté satisfecho. Para indicar que debe ser fácil de adaptar, se ha anotado el proceso completo con el atributo de calidad *Modificabilidad*. Por su parte, para reflejar que la comprobación de pedidos tiene que ser eficiente, se han anotado las ramas relativas a la comprobación de pedidos y búsqueda de productos adicionales con este atributo. Esta segunda rama también se ha anotado porque, para que el cliente realmente perciba la eficiencia, es necesario que las actividades comprendidas por ambas ramas cumplan esta restricción, ya que ambas se mezclan antes de enviar la respuesta al cliente. La identificación y

documentación de estas relaciones durante la fase de análisis del negocio puede llegar a ser relativamente simple pero, a medida que avanza el desarrollo, su identificación se hace más complicada. Con las anotaciones definidas, estas relaciones quedan documentadas desde el inicio para que puedan ser reutilizadas.

Una vez documentadas estas relaciones, para simplificar la derivación del modelo de requisitos, se deben identificar qué funcionalidades del sistema darán soporte a las distintas tareas de negocio. Para facilitar esta actividad, se ha desarrollado un algoritmo que guía en la identificación de qué tareas de negocio deben ser cubiertas por cada funcionalidad. Dicho algoritmo, basado en la regla *Coffe-break* de Cockburn [30], identifica conjuntos de tareas de negocio que pueden ser realizadas por un actor sin interrupción. Se puede obtener más información sobre dicho algoritmo en [31].

A medida que se identifican, estas funcionalidades son documentadas sobre los procesos. Para ello, todas las tareas de negocio que son soportadas por una misma funcionalidad son agrupadas en *Casos de Uso de Negocio* (*BUC*, *Business Use Case*). Actualmente este elemento no puede ser modelado nativamente con BPMN. Por consiguiente, se ha extendido el artefacto *Group* con un nuevo elemento para modelar los BUCs, como se muestra en la Fig. 3.

En la Fig. 4 se puede observar que en el proceso de la tienda online se han identificado y anotado seis BUCs. Cada uno de ellos cubre una funcionalidad del sistema, como pueden ser la realización, validación o comprobación de pedidos. El modelado de los BUCs, además de facilitar la derivación de los requisitos del sistema (como se muestra en la siguiente sec-

¹En este artículo solo se muestra la parte más importante para el ejemplo. En la última sección se indica dónde visualizar la extensión completa.

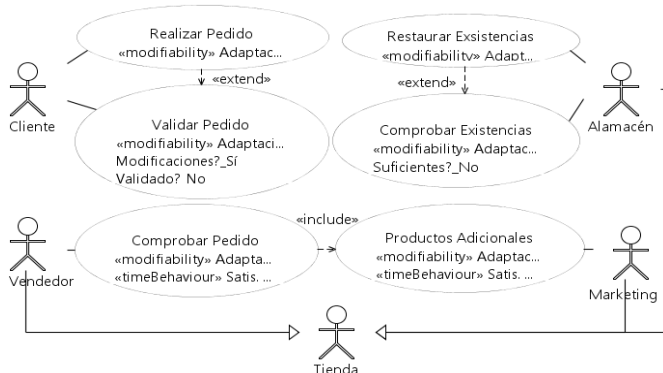


Figura 5. Diagrama de Casos de Uso extraído del proceso de negocio gestión de pedidos.

ción), proporciona al ingeniero una primera aproximación de las funcionalidades del sistema, lo que favorece su discusión con los expertos del negocio.

B. Derivación de los Requisitos y sus Relaciones

Los artefactos generados a partir del análisis del negocio son utilizados como base para definir los requisitos funcionales y no-funcionales del sistema. Así, los procesos de negocio anotados son reutilizados para derivar los requisitos funcionales y sus relaciones con los atributos de calidad.

A partir de las anotaciones de los BUCs se pueden derivar los Casos de Uso del sistema. Además, la ordenación de los BUCs con respecto al flujo del proceso denota la existencia de relaciones entre Casos de Uso. En la Fig. 5 se muestran los Casos de Uso derivados del proceso de gestión de pedidos junto con las relaciones entre Casos de Uso. Como se puede observar, se ha derivado un actor por cada *lane* o *pool* del proceso y un Caso de Uso por cada BUC. Además, se han identificado tres relaciones, una de tipo *include* y dos de tipo *extend*.

Para facilitar y automatizar la identificación de dichas relaciones, se han definido una serie de patrones que detallan las situaciones más comunes de la ordenación de los BUCs con respecto al flujo del proceso que dan lugar a cada tipo de relación. Así, la relación de tipo *include* entre los Casos de Uso *Comprobar Pedido* y *Productos Adicionales* ha sido identificada porque, para que el primer Caso de Uso pueda continuar después del *gateway* paralelo, es necesario haber completado las acciones comprendidas por el segundo. Además, se han definido transformaciones ATL que automatizan la derivación de los Casos de Uso y la aplicación de los patrones. Dichos patrones y reglas ATL se encuentran detallados en [31].

Una vez generado el diagrama de Casos de Uso, se pueden derivar las relaciones entre requisitos funcionales y atributos de calidad. Estas relaciones no se pueden modelar nativamente en estos diagramas. Para que esto sea posible, se ha definido un perfil con un conjunto de estereotipos que extienden la metaclass *ExtensionPoint*. Se ha extendido esta metaclass para que, de esta forma, se mantenga la legibilidad del diagrama cuando se modela un gran número de relaciones.

Los estereotipos creados han sido definidos teniendo en cuenta el modelo de calidad de la ISO/IEC 25010 [28] y, para

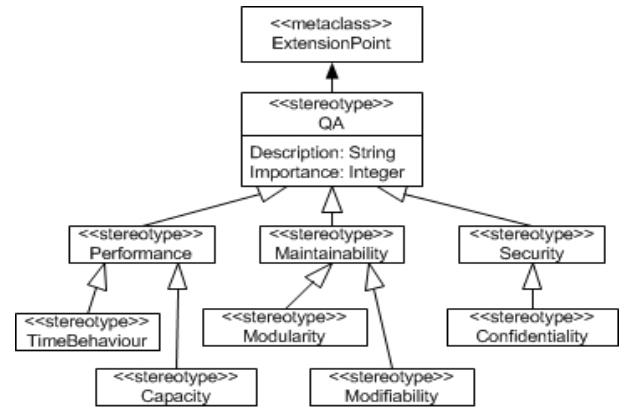


Figura 6. Extracto del perfil para el modelado de relaciones entre atributos de calidad y Casos de Uso.

su modelado, se ha utilizado el plugin de Eclipse Papyrus [32]. La Fig. 6 muestra los estereotipos más importantes para el caso de ejemplo. Estos estereotipos han sido aplicados en el diagrama de Casos de Uso de la tienda online de la Fig. 5. Como se puede observar, todos los Casos de Uso han sido anotados con el estereotipo *Modificabilidad*, ya que el proceso completo debe ser fácil de adaptar, y los Casos de Uso *Comprobar Pedido* y *Productos Adicionales* han sido anotados con el estereotipo *Comportamiento en el Tiempo*, ya que es necesario que se ejecuten en menos de 0.7 segundos para que el cliente esté satisfecho.

Para automatizar la derivación de estas relaciones a partir de los procesos de negocio anotados, se han definido transformaciones ATL. Estas reglas, cada vez que se deriva un Caso de Uso a partir de un BUC, comprueban si dicho BUC estaba restringido por algún atributo de calidad. En caso afirmativo, identifican qué tipo de atributo es y lo anotan en el Caso de Uso derivado. A continuación se muestra un extracto de una de estas reglas. En concreto, la regla presentada es la encargada de anotar el Caso de Uso con el atributo de calidad adecuado. Para ello, crea un nuevo punto de extensión en el Caso de Uso y aplica el estereotipo correspondiente.

```

lazy rule setQAUCRelationship {
  from
    uc : outMM!UseCase,
    cValue : inMM!CategoryValue
  to
    extensionPoint : outMM!ExtensionPoint (
      name <- cValue.value,
      useCase <- uc )

  do{
    extensionPoint.applyStereotype(
      profile!Stereotype.allInstances()->
      any( e | e.name = cValue.getName()));
  }
}

```

Los estereotipos y las reglas ATL definidas permiten generar automáticamente un diagrama de Casos de Uso del sistema alineado con la especificación del negocio. Además, estos

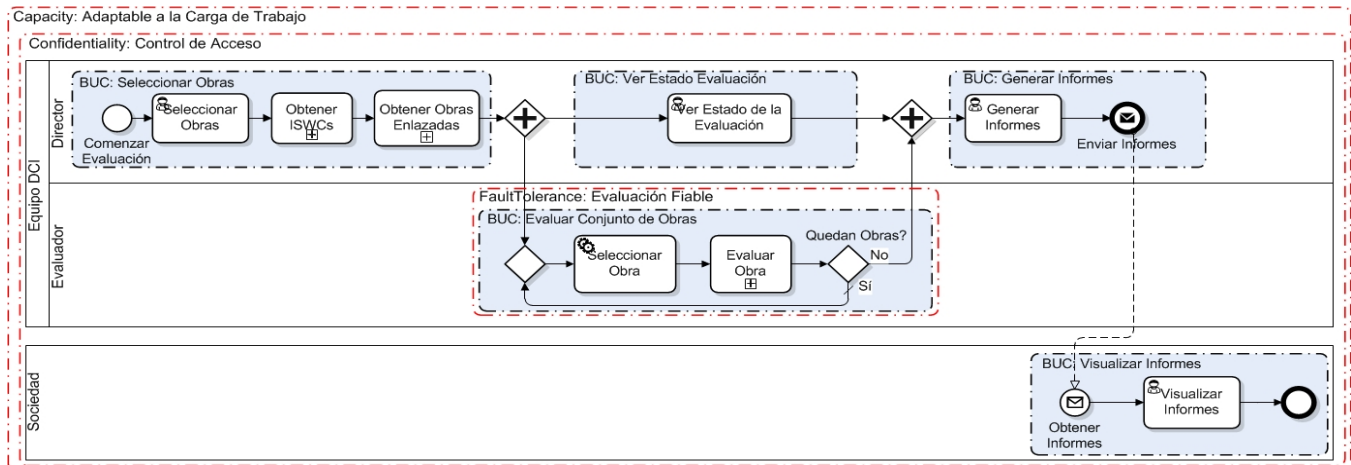


Figura 7. Proceso de negocio para la evaluación de obras musicales.

estereotipos facilitan la documentación y visualización de las interdependencias entre requisitos funcionales y no funcionales, permitiendo que dichas relaciones sean analizadas en futuras fases del desarrollo. En concreto, las anotaciones definidas están especialmente orientadas a que sean reutilizadas por herramientas que asisten al arquitecto en la toma de decisiones arquitectónicas.

IV. VALIDACIÓN DE LA PROPUESTA

Este trabajo ha sido validado en el proyecto *Data Cleanup & Integrity (DCI)* desarrollado por la empresa Gloin². El objetivo principal de este proyecto era desarrollar un producto software que permitiera medir periódicamente la calidad de la información publicada por distintas sociedades de gestión de derechos autor. En concreto, se debía medir la calidad de los datos de más de sesenta millones de obras musicales publicadas por unas cuarenta sociedades. Este producto debía facilitar la medición de dos dimensiones de la calidad de datos: la integridad y la consistencia de la información [33]. Con la integridad se comprueba que los datos de las obras musicales sean válidos, según una serie de reglas del negocio. Con la consistencia se comprueba que los datos de las obras pertenecientes a cada sociedad estén correctos en el resto de sociedades.

Para desarrollar este producto se hizo un exhaustivo análisis del negocio para identificar los objetivos y los procesos a los que el nuevo sistema debía dar soporte. Se identificaron tres objetivos de negocio principales:

- Los procesos de negocio debían ser fácil de adaptar y escalar según la carga de trabajo, puesto que el número de obras evaluables varía según las modificaciones y actualizaciones de cada sociedad.
- Control exhaustivo del acceso a los datos, ya que los datos de las obras son estrictamente confidenciales.
- Los procesos de extracción y posterior evaluación de las obras debían ser fiables y robustos, para que no fallaran, y eficientes, para que se ejecutasen durante el periodo asignado. El proceso de evaluación de la calidad debía

durar como máximo un fin de semana. De esta forma, los sistemas de las sociedades podían absorber la carga de trabajo que supone sin influir en su trabajo diario.

También se identificaron y documentaron los procesos de negocio a los que el sistema debía dar soporte. Los cuatro procesos principales son:

- Envío completo de un repertorio. El equipo encargado de medir la calidad de datos (DCI Team) puede pedir a las sociedades que realicen la extracción de todo su repertorio completo de obras para medir la calidad de sus datos.
- Envío de un repertorio incremental. Una vez extraído el repertorio completo, el DCI Team puede pedir a las sociedades que realicen una extracción incremental de su repertorio, con solamente aquellas obras musicales que son nuevas en la sociedad o que han sido actualizadas.
- Evaluación de la calidad de datos. Se evalúa la integridad y la consistencia de los datos de las obras musicales de las distintas sociedades.
- Evaluación de los datos de una obra. Cada sociedad debe poder evaluar la calidad de los datos específicos de una determinada obra musical.

La Fig. 7 muestra el proceso de negocio para la evaluación de la calidad de datos. Este proceso, primero, establece las obras a evaluar, obtiene sus ISWCs (código internacional de la obra) y las relaciones con los datos de dichas obras en el resto de sociedades; posteriormente, evalúa la integridad y la consistencia de cada obra y, finalmente, genera los informes de la evaluación y los envía a las distintas sociedades.

Una vez identificada y modelada la información de negocio, se anotaron las relaciones entre procesos de negocio y objetivos de negocio. Se identificó que todos los procesos de negocio debían ser seguros, escalables a la carga de trabajo y que, específicamente las partes relativas a la extracción de obras de las sociedades y a la evaluación de la calidad de dichas obras, debían ser fiables y eficientes. Así, en el proceso de evaluación de obras, Fig. 7, se anotaron tres relaciones: dos indican que todo el sistema debe ser escalable y seguro, y la otra refleja que la evaluación de la consistencia y la integridad

²<http://www.gloin.es>

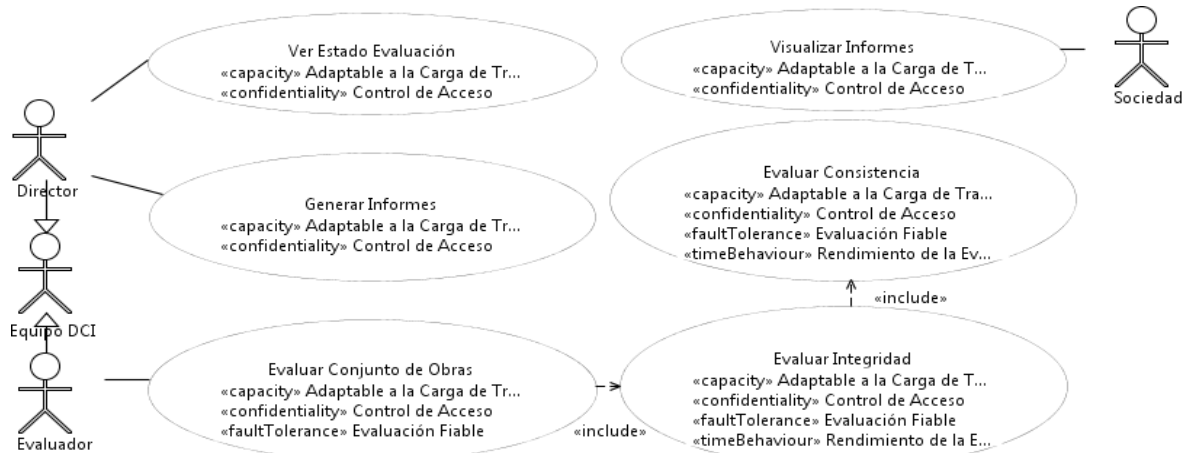


Figura 8. Extracto del diagrama de Casos de Uso extraído del proceso de evaluación de obras musicales.

debe ser tolerante a fallos para que el sistema sea más fiable.

Posteriormente, se identificaron y anotaron los Casos de Uso de negocio necesarios para dar soporte a los procesos. Para los cuatro procesos indicados anteriormente, se identificaron un total de treinta y un BUCs. En el proceso de evaluación de obras, Fig. 7, se anotaron cinco BUCs que cubrían, entre otras funcionalidades, la visualización del estado de la evaluación, la evaluación de las obras o la generación de informes.

Al aplicar a los procesos de negocio anotados las transformaciones ATL presentadas anteriormente y en [31], se obtuvieron automáticamente los diagramas de Casos de Uso del sistema. Estos diagramas estaban compuestos por: ocho actores, treinta y un Casos de Uso, cuarenta y tres relaciones entre Casos de Uso, o entre actores, (nueve de ellas relaciones de tipo *include* o *extend*) y setenta y cinco relaciones entre Casos de Uso y atributos de calidad. La Fig. 8 muestra un extracto del diagrama extraído del proceso de evaluación de obras musicales. Dicho diagrama muestra seis Casos de Uso, cuatro de ellos identificados como BUCs en el proceso de la Fig. 7 y otros dos extraídos del sub-proceso *Evaluar Obra* (incluido en el proceso anterior). Además, se puede observar que, por ejemplo, para el Caso de Uso *Evaluar un conjunto de obras* también se han derivado las restricciones de adaptación dependiendo de la carga de trabajo, de seguridad y de tolerancia a fallos.

Los modelos de requisitos generados fueron utilizados para generar la arquitectura del sistema. Para conseguir la robustez y fiabilidad se decidió desplegar el sistema en un entorno Cloud [34]. Para mejorar la robustez y el rendimiento, las funcionalidades específicas anotadas con estos requisitos (como puede ser *Evaluar un conjunto de obras*, como se muestra en la Fig. 8) fueron desplegadas en entornos Cloud específicos.

Para hacer el sistema seguro, el arquitecto decidió aplicar el patrón *authorization* [35] sobre todos los requisitos funcionales anotados con el atributo de calidad de confidencialidad. En un nivel más detallado de la arquitectura, dicho patrón fue soportado por el framework de seguridad Spring Security [36].

Finalmente, una vez evaluados los Casos de Uso restringidos por el atributo de calidad de adaptación del sistema a la carga de trabajo, se decidió modularizar el sistema en una serie de agentes, o subsistemas, que se comunican mediante JMS [37]. Cada uno de estos agentes tenía una funcionalidad específica: obtención de datos de las obras, almacenamiento de los datos, evaluación de la calidad de datos o generación de estadísticas. Así, en tiempo de ejecución, durante la extracción o evaluación de las obras, se podían arrancar o parar tantos agentes como se desearan dependiendo de la carga de trabajo y el rendimiento deseado. En [38] se puede encontrar un extracto de la arquitectura detallada por uno de los arquitectos que participó en el proyecto. Como se ha podido observar, gracias al modelado de las interdependencias entre elementos del negocio y entre requisitos, se pudieron tomar decisiones arquitectónicas para diseñar y desarrollar un sistema alineado con el negocio.

V. DISCUSIÓN

En el proyecto DCI han estado trabajando cuatro expertos del negocio de los derechos de autor, un jefe de proyectos, un ingeniero de requisitos, dos arquitectos software y cuatro desarrolladores. Con el objetivo de obtener información sobre el uso de la propuesta presentada en este artículo, se realizó un análisis cualitativo. Para ello, se realizaron entrevistas a dos expertos del negocio, al ingeniero de requisitos y a los dos arquitectos software. No se realizaron entrevistas al jefe de proyectos ni a los desarrolladores porque la propuesta presentada no influye directamente sobre su método de trabajo.

Para realizar este análisis cualitativo se siguió la metodología detallada por Yin en [39]. Se realizaron entrevistas a mitad del proyecto y al final. Estas entrevistas consistían en preguntas abiertas con los siguientes objetivos:

- Identificar si la propuesta y las herramientas que la soportan funcionaron correctamente.
- Obtener información sobre la experiencia de uso de la propuesta y de las herramientas.
- Obtener sugerencias y mejoras que podrían ser incorporadas al trabajo.

Las entrevistas realizadas a mitad del proyecto proporcionaron información sobre el uso de la propuesta por los expertos de negocio y por el ingeniero de requisitos, ya que los arquitectos llevaban poco tiempo involucrados en el proyecto. En estas entrevistas se destacó que la propuesta facilitaba la discusión sobre cómo los objetivos de negocio influyen sobre cada parte de los procesos de negocio, mejorando también su documentación. Así mismo, el ingeniero resaltó que la generación automática del modelo de requisitos anotado reducía el tiempo de documentación y facilitaba la generación de un conjunto de requisitos alineados con el negocio. Como sugerencia, el ingeniero de requisitos indicó que sería de gran utilidad que las herramientas facilitaran el refinamiento de los requisitos, por ejemplo, generando diagramas de actividad a partir de los Casos de Uso. De esta forma sería más fácil mantener la alineación hasta la fase de diseño del sistema.

Las entrevistas realizadas al final del proyecto proporcionaron información sobre el uso de la propuesta por los arquitectos. Éstos destacaron que las anotaciones de las interdependencias les ayudaron a conocer con menor esfuerzo la importancia de cada requisito, ya que podían visualizar, de forma más precisa, qué funcionalidades estaban restringidas por cada atributo de calidad y cómo estaban restringidas. Resaltaron que esta información les facilitó la toma de decisiones sobre qué patrones arquitectónicos aplicar y dónde aplicarlos. Como limitación actual del sistema, los arquitectos sugirieron que sería deseable que las relaciones anotadas en los requisitos también pudieran ser derivadas al diagrama de clases. De esta forma podrían conocer exactamente sobre qué clases aplicar los patrones arquitectónicos necesarios para facilitar el cumplimiento de los atributos de calidad.

El uso de la propuesta en un proyecto real nos ha permitido conocer sus fortalezas y debilidades. La metodología y las herramientas desarrolladas ayudan a discutir y documentar las relaciones entre elementos del negocio y entre requisitos. Esta información, a su vez, ayuda al arquitecto a realizar un diseño correctamente alineado con los objetivos y procesos del negocio. Sin embargo, para aumentar la utilidad de dichas relaciones sería deseable que también se diese soporte a su especificación y refinamiento en otros documentos y modelos.

Una vez terminada la validación de la propuesta, se ha trabajado en mejorar sus debilidades. Actualmente se están evaluando técnicas para derivar los diagramas de actividad, de cada Caso de Uso, a partir de los procesos de negocio anotados. Además, se está trabajando en derivar un modelo de dominio inicial a partir de los datos de entrada y salida de las tareas de los procesos de negocio.

VI. CONCLUSIONES

Desde las primeras fases del desarrollo se pueden identificar interdependencias entre elementos del negocio o entre requisitos. Debido a que estas fases se centran en detallar cada uno de los elementos del negocio o cada uno de los requisitos, las relaciones entre ellos quedan documentadas de forma implícita. Sin embargo, estas relaciones son especialmente importantes durante el diseño del sistema. Son necesarias para

conocer desde la importancia de los distintos atributos de calidad hasta para evaluar el impacto que puede tener una determinada decisión arquitectónica.

En este artículo se han presentado una serie de extensiones, para BPMN 2 y para Diagramas de Casos de Uso de UML 2, y un conjunto de transformaciones ATL que permiten documentar y derivar automáticamente dichas relaciones. Esta propuesta ha sido validada en un proyecto real. Esto nos ha permitido ratificar que la documentación y tratamiento de estas interdependencias permite al arquitecto realizar diseños mejor alineados con el negocio. También nos ha permitido identificar distintos puntos en los que perfeccionar la propuesta.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación de España dentro del proyecto TIN2012-34945, por la Consejería de Empleo, Empresa e Innovación del Gobierno de Extremadura dentro del proyecto GR10129, y por el Fondo Europeo de Desarrollo Regional (FEDER).

MATERIAL ADICIONAL

En la página web <http://ieee.zentipede.org> se ha incluido material adicional relacionado con el trabajo presentado en el artículo. Este material incluye los diagramas y ficheros XML y XMI generados durante el proceso de ejemplo de la tienda online, las transformaciones ATL, la extensión de BPMN y el perfil de UML completo.

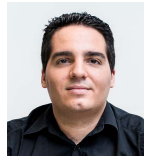
REFERENCIAS

- [1] E. Cardoso, J. Almeida, and G. Guizzardi, "Requirements engineering based on business process models: A case study," in *Enterprise Distributed Object Computing Conference Workshops*, 2009. EDOCW 2009. 13th, 2009, pp. 320–327.
- [2] H. Traetteberg and J. Krogstie, "Enhancing the usability of bpm-solutions by combining process and user-interface modelling," in *The Practice of Enterprise Modeling*, J. Stirna, A. Persson, W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, and C. Szyperski, Eds. Springer Berlin Heidelberg, 2009, vol. 15, pp. 86–97.
- [3] S. Gao and J. Krogstie, "A combined framework for development of business process support systems," in *The Practice of Enterprise Modeling*, A. Persson, J. Stirna, W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, and C. Szyperski, Eds. Springer, 2009, vol. 39, pp. 115–129.
- [4] A. Ullah and R. Lai, "Modeling business goal for business/it alignment using requirements engineering," *Journal of Computer Information Systems*, vol. 51, no. 3, pp. 21–28, 2011.
- [5] G. Kotonya and I. Sommerville, *Requirements engineering: processes and techniques*, ser. *Worldwide series in computer science*. J. Wiley, 1998.
- [6] OpenUP, 2013. [Online]. Available: <http://http://epf.eclipse.org/wikis/openup/>
- [7] P. Avgeriou, J. Grundy, J. G. Hall, P. Lago, and I. Mistrík, Eds., *Relating Software Requirements and Architectures*. Springer, 2011.
- [8] R. Capilla, M. Ali Babar, and O. Pastor, "Quality requirements engineering for systems and software architecting: methods, approaches, and tools," *Requirements Engineering*, vol. 17, no. 4, pp. 255–258, 2012.
- [9] A. G. Dahlstedt and A. Persson, "Requirements interdependencies moulding the state of research into a research agenda," in *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2003)*, 2003, pp. 71–80.
- [10] L. Xu, H. Ziv, and D. Richardson, "Towards modeling nonfunctional requirements in software architecture" In *Proceedings of Aspect-Oriented Software Design, Workshop on Aspect Oriented Requirements Engineering and Architecture Design*, 2005.

- [11] BPMN, "Business Process Modeling Notation 2.0." [Online]. Available: <http://www.bpmn.org/>
- [12] ATLAS team, "ATLAS Transformation Language (ATL)." [Online]. Available: <http://www.eclipse.org/atl/>
- [13] J. Berrocal, J. García-Alonso, and J. M. Murillo, "Facilitating the selection of architectural patterns by means of a marked requirements model," in ECSCA, ser. LNCS, M. A. Babar and I. Gorton, Eds., vol. 6285. Springer, 2010, pp. 384–391.
- [14] J. García-Alonso, J. Berrocal, and J. M. Murillo, "Architectural decisions in the development of multi-layer applications," in ICSEA: Fifth International Conference on Advances in System Testing and Validation Lifecycle, 2013.
- [15] T. Mitra, "Business-driven development, last access june 2012," 2005. [Online]. Available: <http://www.ibm.com/developerworks/webservices/library/ws-bdd/>
- [16] BMM, "Business Motivation Model." [Online]. Available: <http://www.omg.org/spec/BMM/>
- [17] E. S.-K. Yu, "Modelling strategic relationships for process reengineering," Ph.D. dissertation, University of Toronto, Toronto, Ont., Canada, 1995.
- [18] T. Allweyer, BPMN 2.0: Introduction to the Standard for Business Process Modeling. Books on Demand GmbH, 2010.
- [19] F. Aburub, M. Odeh, and I. Beeson, "Modelling non-functional requirements of business processes," *Inf. Softw. Technol.*, vol. 49, pp. 1162–1171, 2007.
- [20] J. Bocanegra, J. Pena, and A. Ruiz, "Interorganizational business modeling: an approach for traceability of goals, organizational models and business processes," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 9, no. 1, pp. 847–854, 2011.
- [21] C. J. Pavlovski and J. Zou, "Non-functional requirements in business process modeling," in *Proceedings of the fifth Asia-Pacific conference on Conceptual Modelling - Volume 79*. Darlinghurst, Australia: Australian Computer Society, Inc., 2008, pp. 103–112.
- [22] R. M. Dijkman and S. M. M. Joosten, "Deriving use case diagrams from business process models," University of Twente, Tech. Rep. TR-CTIT-02-08, 2002.
- [23] A. Rodriguez, E. Fernandez-Medina, M. Piattini, and J. Trujillo, "Secure business processes defined through a uml 2.0 extension," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 6, no. 4, pp. 339–346, 2008.
- [24] J. de la Vara and J. Sánchez, "Bpmn-based specification of task descriptions: Approach and lessons learnt," in *Requirements Engineering: Foundation for Software Quality*, M. Glinz and P. Heymans, Eds. Berlin, Heidelberg: Springer, 2009, vol. 5512, pp. 124–138.
- [25] F. Siqueira and P. Silva, "Transforming an enterprise model into a use case model using existing heuristics," in *Model-Driven Requirements Engineering Workshop*, 2011, pp. 21–30.
- [26] J. Dörr, "Elicitation of a complete set of non-functional requirements." Ph.D. dissertation, University of Kaiserslautern, 2011.
- [27] J. M. Verner, K. Cox, S. J. Bleistein, and N. Cerpa, "Requirements engineering and software project success: an industrial survey in australia and the u.s." *Australasian J. of Inf. Systems*, vol. 13, no. 1, 2005.
- [28] International Standard Organization (ISO/IEC), "ISO/IEC 25010:2011- Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models," ISO/IEC, Tech. Rep., 2011.
- [29] "Eclipse BPMN2 Modeler." [Online]. Available: <http://eclipse.org/bpmn2-modeler/>
- [30] A. Cockburn, *Writing Effective Use Cases*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.
- [31] J. Berrocal, J. García-Alonso, C. Vicente-Chicote, and J. M. Murillo, "A pattern-based and model-driven approach for deriving it system functional models from annotated business models," in *22nd International Conference on Information Systems Development*. Springer, 2013.
- [32] Eclipse Papyrus. [Online]. Available: <http://www.eclipse.org/papyrus/>
- [33] D. Mcgilvray, *Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information (TM)*. Burlington, MA: Elsevier, 2008.
- [34] "National Institute of Standards and Technology, 2011. [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [35] E. B. Fernandez, "Layers and non-functional patterns," in *In: Proc. Of ChiliPLoP 2003*, 2003.
- [36] Spring Security. [Online]. Available: <http://docs.spring.io/spring-security/site/index.html>
- [37] JavaTM Message Service (JMS). [Online]. Available: <http://jcp.org/aboutJava/communityprocess/final/jsr914/index.html>
- [38] J. Guillén, J. Miranda, J. Murillo, and C. Canal, "A uml profile for modeling multicloud applications," in *Service-Oriented and Cloud Computing*, ser. Lecture Notes in Computer Science, K.-K. Lau, W. Lamersdorf, and E. Pimentel, Eds. Springer Berlin Heidelberg, 2013, vol. 8135, pp. 180–187.
- [39] R. K. Yin, *Case Study Research: Design and Methods*, 3rd Edition (Applied Social Research Methods, Vol. 5), 3rd ed. SAGE Publications, Inc, 2002.
- [40] M. Solinas and L. Antonelli, "Software evolution and design patterns," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 11, no. 1, pp. 347–352, 2013.



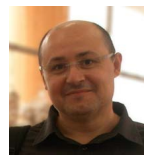
Javier Berrocal is a cofounder of Gloin and a PhD candidate and an assistant professor in the Department of Informatics and Telematic System Engineering at the University of Extremadura (Spain), where he belongs to the Quercus Software Engineering Group (QSEG). Berrocal received an MSc in software engineering from the University of Extremadura in 2008. His current research interests include requirements engineering, software architectures, model-driven development, and business process management.



Jose Garcia Alonso is a cofounder of Gloin and a PhD candidate and an assistant professor in the Department of Informatics and Telematic System Engineering at the University of Extremadura (Spain), where he belongs to the Quercus Software Engineering Group (QSEG). He received an MSc in software engineering from the University of Extremadura in 2009. His research interests include software product lines, software architectures, model-driven development, and framework development.



Cristina Vicente Chicote is an associate professor in the Department of Informatics and Telematic System Engineering at the University of Extremadura (Spain), where she belongs to the Quercus Software Engineering Group (QSEG). She received a BSc in Computer Science at the University of Murcia (Spain) in 1997, and a PhD degree (European Mention) at the Technical University of Cartagena (Spain) in 2005. Her current research interests include model-driven engineering, component-based software development, software product lines, and (self-) adaptive system development.



Juan Manuel Murillo is a cofounder of Gloin and an associate professor in the Department of Informatics and Telematic System Engineering at the University of Extremadura (Spain), where he belongs to the Quercus Software Engineering Group (QSEG). Murillo received a PhD in computer science at the University of Extremadura (Spain) in 2001. His current research interests include software architectures, mobile computing, and cloud computing.