

Vicente Pelechano Ferragud,
Joan Josep Fons Cors, Pau
Giner Blasco

Centro de Investigación en Métodos de Producción de Software (ProS), Universidad Politécnica de Valencia

<{pele,jjfons,pginer}@pros.upv.es>

1. Introducción

Las Tecnologías de la Información (TI) permiten crear un mundo digital donde la información se puede procesar de forma automática mejorando la eficiencia de los Sistemas de Información (SI). Sin embargo, los ordenadores todavía poseen una visión limitada del mundo real que están gestionando. Así pues, todavía existe un reto importante que consiste en automatizar el enlace entre el mundo real y el mundo físico.

Actualmente, los SI que tratan con objetos del mundo real (como maletas en un aeropuerto o productos de un supermercado) normalmente son informados por humanos. Este uso de los humanos como portadores de información resulta ineficiente y propenso a errores. La distancia entre el mundo físico y el mundo digital acaba comúnmente en equipajes perdidos o largas colas en los supermercados.

La visión que ofrece la Internet de las cosas [1] pretende reducir esta distancia para hacer más fluidas las actividades cotidianas. Proporcionando una identidad digital a los objetos del mundo real, los SI pueden manejar estos objetos de forma automática. Esto permite que los objetos físicos participen de forma activa en los procesos de negocio reduciendo la distancia entre los mundos físicos y virtuales [2]. Además, la amplia disponibilidad de dispositivos móviles y teléfonos inteligentes con capacidades avanzadas permiten a los usuarios acceder a la información y a los servicios donde ellos los necesitan.

Aunque el desarrollo de este tipo de sistemas es factible, la heterogeneidad tecnológica existente en el ámbito de la Identificación Automática (Auto-ID) y los requisitos altamente cambiantes de los procesos de negocio dificultan su construcción, mantenimiento y evolución. Así pues, existe una necesidad de pasar de soluciones ad-hoc a métodos de desarrollo sólidos que garanticen la calidad del producto final. Las técnicas de la Ingeniería Dirigida por Modelos [3] pueden ayudar a los desarrolladores a proporcionar métodos de desarrollo para la construcción de este tipo de sistemas.

El principal objetivo de este trabajo es mostrar cómo el Desarrollo Dirigido por Modelos (DDM) puede ayudar a sistematizar el

Desarrollo Dirigido por Modelos aplicado a la Internet de las cosas

Resumen: La visión que ofrece la Internet de las cosas pretende reducir la distancia entre el mundo físico y el mundo digital para llegar a hacer más fluidas las actividades cotidianas. Los Sistemas de Información pueden manejar de forma automática los objetos del mundo real si se les proporciona una identidad digital. Esto permite que los objetos físicos participen de forma activa en los procesos de negocio. En este tipo de sistemas, la heterogeneidad tecnológica existente en el ámbito de la Identificación Automática (Auto-ID) y los requisitos altamente cambiantes de los procesos de negocio dificultan su construcción, mantenimiento y evolución. Este trabajo muestra cómo el Desarrollo Dirigido por Modelos puede ayudar a sistematizar el desarrollo de sistemas software de soporte a los procesos de negocio que integren elementos físicos.

Palabras clave: Desarrollo Dirigido por Modelos, identificación automática, interacción implícita, Internet de las cosas, modelos, procesos de negocio, workflows móviles.

Autores

Vicente Pelechano Ferragud es profesor Titular de Universidad del Centro de Investigación en Métodos de Producción de Software en la Universidad Politécnica de Valencia (UPV). Sus intereses en investigación incluyen el Desarrollo Dirigido por Modelos, la Ingeniería Web, la Computación Ubicua y Móvil, y el Modelado de Procesos de Negocio, entre otras. Es doctor en Informática por la UPV y actualmente lidera la supervisión técnica del proyecto MOSKitt.

Joan Josep Fons Cors es profesor Contratado Doctor en el Centro de Investigación en Métodos de Producción de Software en la Universidad Politécnica de Valencia (UPV). Sus intereses en investigación incluyen el Desarrollo Dirigido por Modelos, la Ingeniería Web, la Computación Ubicua y Autónoma, el Desarrollo de Interfaces de Usuario y la Ingeniería de Líneas de Producto. Es doctor en Informática por la UPV.

Pau Giner Blasco es consultor en Everis y miembro del Centro de Investigación en Métodos de Producción de Software en la Universidad Politécnica de Valencia (UPV). Sus intereses en investigación se centran en el modelado de procesos de negocio, el Internet de las Cosas y las aplicaciones móviles. Es doctor en informática por la UPV.

desarrollo de sistemas software de soporte a los procesos de negocio que integren elementos físicos. El proceso de desarrollo definido se centra en los requisitos específicos del enlace entre los mundos físicos y virtuales. Para la especificación del sistema se ha definido un lenguaje de modelado para manejar los requisitos específicos del dominio de la Internet de las cosas. A partir de estos requisitos se obtiene una solución software siguiendo un conjunto sistemático de pasos. Esta solución se soporta a través de un *framework* arquitectónico diseñado específicamente para dar soporte a los requisitos de la Internet de las cosas y para sobrevivir a la continua evolución de tecnologías.

Concretamente, nuestro tipo de aplicaciones destino son los flujos de trabajo móviles que integran elementos físicos. Estas aplicaciones son procesos de negocio que aprovechan las capacidades de los dispositivos móviles

para la identificación de elementos físicos. La elevada heterogeneidad existente en las tecnologías de identificación, la fragmentación de las plataformas móviles y la naturaleza tan cambiante de los procesos de negocio hace que sea complicado desarrollar este tipo de sistemas de una manera sólida. En este trabajo se aplican técnicas de modelado para abordar el desarrollo de este tipo de sistemas desde un nivel alto de abstracción.

Lo que resta de este trabajo se estructura como sigue a continuación. La **sección 2** presenta las principales ideas para llevar a cabo la integración de los procesos de negocio con los principios de la Internet de las cosas. En la **sección 3** se discute por qué adoptar una estrategia de modelado para ayudar en la definición de este tipo de sistemas de forma que se evite centrarse en aspectos demasiado técnicos. La **sección 4** presenta las principales ideas de método de diseño que dé soporte

te al desarrollo de sistemas en el ámbito de la Internet de las cosas siguiendo una estrategia dirigida por modelos. En la **sección 5** se presentan las herramientas que dan soporte a la propuesta metodológica. Finalmente, en la **sección 6** se presentan las conclusiones del trabajo.

2. Procesos de negocio y la Internet de las cosas

La integración de objetos del mundo real en los procesos de negocio se está llevando a cabo de manera exitosa reduciendo la brecha entre medios, errores humanos y problemas de retardo en la información [2]. Se han obtenido muchos beneficios en términos económicos [4] y de mejora de procesos [5][6]. Conseguir una mejor integración de los mundos virtuales y reales no sólo mejora los procesos de negocio sino que también facilita el desarrollo de nuevos modelos de negocio [7][8].

Sin embargo, abordar el desarrollo de este tipo de sistemas no es una tarea fácil. Los procesos de negocio están cambiando constantemente, lo que a su vez requiere la evolución de los SI que los soportan. Además, los sistemas software en el contexto de la Internet de las cosas involucran una gran variedad de tecnologías para conseguir desarrollar el puente entre los mundos físicos y virtuales. Esta heterogeneidad obliga al desarrollador a conocer los detalles de cada una de las tecnologías necesarias para desarrollar el sistema, haciendo que estos sistemas sean difíciles de desarrollar y mantener. Desde el punto de vista metodológico, existe la necesidad de un método de desarrollo sistemático que pueda liberar a los desarrolladores de conocer los detalles tecnológicos, y que a su vez permita una rápida propagación de los cambios en los requisitos a las soluciones tecnológicas.

Este trabajo presenta un método que proporciona un mecanismo para definir el grado deseado de automatización del enlace físico-virtual para un determinado proceso de nego-

cio. Con el objetivo de sistematizar el desarrollo de este tipo de sistemas, el método se basa en los principios de la iniciativa *Business Process Management* (BPM). BPM es una iniciativa que promueve la reingeniería continua de procesos de negocio. Las soluciones actuales en el ámbito de BPM están enfocadas principalmente en el mundo digital (por ejemplo, la orquestación de servicios digitales) por lo que no existe soporte para tratar con las particularidades del enlace físico-virtual en las diferentes etapas del ciclo de desarrollo propuesto por BPM. Este trabajo se construye sobre técnicas existentes en BPM y las extiende para integrar los procesos de negocio con el mundo físico a diferentes niveles de abstracción. Las técnicas BPM existentes se complementan con el soporte para la captura de requisitos de identificación, para la evaluación de la participación del usuario en un entorno real, y para la ejecución de flujos de trabajo en un plataforma software.

3. ¿ Por qué un enfoque de modelado?

Tradicionalmente, la aplicación de *identificación automática* (Auto-ID) a procesos de negocio se ha abordado principalmente desde una perspectiva puramente tecnológica (desarrollando *middleware* de integración y *frameworks* arquitectónicos). Sin embargo, la implantación de sistemas con soporte a Auto-Id implica mucho más que comprar las etiquetas adecuadas e instalar los lectores adecuados [9].

La forma en la que un objetivo de negocio se consigue depende de las propiedades de la integración físico-virtual. Ciertos modelos de negocio sólo son factibles con un nivel adecuado de automatización del enlace físico-virtual [7]. Por ejemplo, usar tecnología RFID para identificar productos en un supermercado permite automatizar el proceso de pago [10], y no requiere de la participación de un cajero en el proceso. Así, cuando se modela un proceso de negocio no es posible determinar qué tareas requieren manejar elementos

físicos (por ej. que requieran un cajero para llevarlas a cabo o no) si no existe forma de explicitar cómo deben participar en el proceso. Los modelos en nuestro enfoque son la clave para proporcionar esta información, enlazando los requisitos de identificación con los tecnológicos de forma gradual.

La abstracción es uno de los principios fundamentales de la ingeniería del software para gestionar la complejidad [11]. Nuestro enfoque hace uso de técnicas de modelado para promover la abstracción en el desarrollo de flujos de trabajo móviles que integran elementos físicos. Abstrayendo los detalles técnicos, podemos describir la conexión física-virtual de un flujo de trabajo con independencia de la tecnología concreta usada para la implementación. En el caso de flujos de trabajo móviles que integran elementos físicos, las técnicas de modelado se aplicarán para obtener los siguientes beneficios:

- **Centrarse en el proceso.** Nuestro enfoque promueve la separación de aspectos para permitir a los diseñadores centrarse en un aspecto específico del flujo de trabajo a la vez. Los analistas del negocio pueden definir la forma en la que los elementos físicos participan en un proceso de negocio sin pensar en las limitaciones de la tecnología. Ellos pueden pensar en la forma en la que fluyen las tareas del proceso, y más tarde en los mecanismos de identificación apropiados que se elegirán para llevar a cabo sus requisitos.

- **Explorar el espacio de la solución.** El uso de modelos permite capturar, no sólo la solución específica sino el fundamento que hay detrás de ella. De esta forma, se pueden reconsiderar soluciones alternativas y el conocimiento de diseño se puede reutilizar para abordar problemas similares. Además, el soporte a la trazabilidad permite identificar más fácilmente los elementos del modelo afectados cuando se detectan problemas durante la evaluación del sistema.

- **Soportar la evolución del sistema.** La naturaleza rápidamente cambiante de los procesos de negocio y la heterogeneidad de las tecnologías de identificación sugiere que los sistemas en esta área de aplicación deben diseñarse para poder evolucionar. Analizando el conocimiento capturado en los modelos se puede determinar de forma sencilla si una nueva tecnología encaja bien con los requisitos del enlace físico-virtual de un proceso de negocio dado.

Nuestro enfoque implica manipular los modelos de forma diferente. En la siguiente sección se presenta una visión general de los pasos que proporciona nuestro método.

4. Visión general del método

Esta sección presenta el método de desarrollo descrito en este trabajo. La etapa de diseño es la inicial en nuestro método (ver **figura 1**). Debido a que sigue una aproximación dirigi-

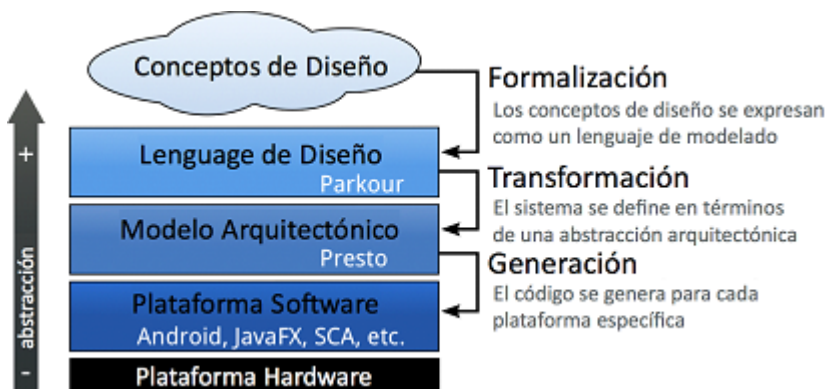


Figura 1. Estrategia para cubrir los niveles de abstracción en el desarrollo de procesos de negocio físicos.

da por modelos, la especificación que se obtiene en esta etapa de diseño es la que guía el resto de etapas en el desarrollo del sistema. Es por ello que la etapa de diseño es la principal del método de desarrollo. En ella, se capturan mediante modelos los conceptos que son relevantes para el desarrollo de procesos de negocio que integran objetos físicos. Las técnicas de modelado describen un sistema mediante abstracciones del espacio del problema. Esto permite expresar el diseño en términos del dominio del problema en vez de términos del espacio de la solución [12]. Al elevar el nivel de abstracción, los sistemas pueden describirse sin tener en cuenta los detalles tecnológicos de la plataforma subyacente. Los modelos se usan para organizar el conocimiento sobre el dominio del problema para guiar el desarrollo. Además, cuando los modelos son procesables y suficientemente precisos, se pueden usar para automatizar la producción del sistema software.

Gracias a las técnicas de DDM, es posible unir los conceptos de alto nivel usados en la etapa de diseño con los detalles técnicos de la plataforma móvil concreta usada para la implementación del sistema.

La **figura 1** muestra como la propuesta permite conectar los conceptos usados en la etapa de diseño (por ej. tarea, nivel de intromisión, interacción física, etc.) con la plataforma concreta de implementación. La propuesta introduce dos niveles para recorrer la distancia que hay entre los niveles de abstracción de los conceptos de diseño y la plataforma software:

■ **Parkour.** Los conceptos de diseño se formalizan en un lenguaje de modelado llamado Parkour. Los diseñadores pueden definir los requisitos del proceso de negocio usando este método de diseño y usar la especificación del sistema durante el desarrollo del sistema. Al formalizar estos conceptos de diseño, las descripciones del sistema pueden procesarse para validar su consistencia y

automatizar algunos pasos del desarrollo. Es posible transformar las especificaciones de los procesos de negocio basadas en Parkour en una solución final usando técnicas DDM.

■ **Presto.** Presto [13] es un *framework* arquitectónico específicamente definido para soportar la ejecución de aplicaciones en el dominio de procesos de negocio físicos. Presto es una arquitectura software sostenible, es decir, una arquitectura que puede evolucionar durante el tiempo por medio de diferentes ciclos tecnológicos. Los elementos de la arquitectura se definen con independencia tecnológica y se usan técnicas de generación de código para transformar la arquitectura genérica en componentes específicos de la plataforma destino elegida.

Creemos que es importante describir el enlace físico-virtual de los procesos de negocio en tiempo de diseño debido a que la manera en que un objetivo de negocio es satisfecho depende de las propiedades de esta integración física-virtual. Nuestra aproximación se basa en el uso de notaciones existentes para modelar procesos de negocio, como la notación *Business Process Modeling Notation* (BPMN). Parkour complementa estas descripciones BPMN con el modelado de tres aspectos que afectan el enlace físico-virtual:

■ **Nivel de intromisión para cada tarea.** Identificamos el nivel de intromisión requerido para cada tarea del proceso de negocio. Cada una puede llevarse a cabo con un nivel de iniciativa y atención diferente, de acuerdo con el marco conceptual introducido en [14].

■ **Uso de técnicas de interacción.** Los usuarios pueden interactuar con los objetos que les rodean de diferentes maneras. Para cada tarea del proceso debe elegirse la técnica de interacción que permita llevar a cabo esta tarea.

■ **Tecnologías de soporte.** Los diferentes dispositivos móviles involucrados en el proceso de negocio deben estar equipados con las tecnologías que soporten las técnicas

de interacción que hayan sido elegidas en la etapa anterior.

Las siguientes secciones proporcionan más detalles sobre estos conceptos. Para ilustrarlos, usaremos el caso de estudio de la Biblioteca Inteligente [13], donde se da soporte a diferentes procesos de negocio en el contexto de una biblioteca.

4.1. Nivel de intromisión para cada tarea

El primer paso del método consiste en detectar las tareas que dan soporte al proceso de negocio y determinar a qué nivel de intromisión deben soportarse. Cada tarea puede soportarse con un nivel diferente de intromisión. El método de diseño que introduce esta sección permite a los diseñadores especificar un proceso de negocio indicando (1) hasta qué punto deben molestar a los usuarios las diferentes tareas, y (2) qué tecnologías pueden usarse para soportar estas interacciones. Debido a que los dispositivos móviles proporcionan mecanismos avanzados de interacción, no consideramos las interacciones como implícitas o explícitas de manera binaria, sino más bien como parte de un continuo. En este trabajo se usa el concepto de nivel de intromisión para indicar hasta qué punto una interacción es implícita o explícita.

Para especificar el nivel de intromisión para una tarea, hacemos uso del marco conceptual presentado en [14]. Este marco define dos dimensiones (ver **figura 2**) para caracterizar las interacciones implícitas: *iniciativa* y *atención*. De acuerdo con el factor iniciativa, una interacción puede ser *reactiva* (el usuario inicia la interacción) o *proactiva* (el sistema toma la iniciativa). Con respecto al factor atención, una interacción puede tomar lugar en *primer plano* (el usuario es totalmente consciente de la interacción) o en *segundo plano* de atención (el usuario no es advertido de la interacción con el sistema).

Consideramos muy útiles los factores de iniciativa y atención como conceptos independientes. En el caso de procesos de negocio físicos, la automatización y percepción del usuario son factores que suelen cambiar independientemente. Por ejemplo, una tarea automatizada (esto es, proactiva en términos de atención) puede requerir que el usuario sea consciente de ella (es decir, en primer plano en términos de atención) o no (en segundo plano de atención), dependiendo de diferentes factores de contexto (por ej. la carga de trabajo del usuario).

La **figura 3** muestra las tareas más representativas del caso de estudio de la Biblioteca Inteligente, y sus niveles de intromisión definidos. El espacio de intromisión en este caso fue definido dividiendo cada eje en diferentes partes. El eje de iniciativa se dividió en dos:



Figura 2. Marco conceptual de interacción implícita usada. Ejemplos de interacciones móviles para cada región.

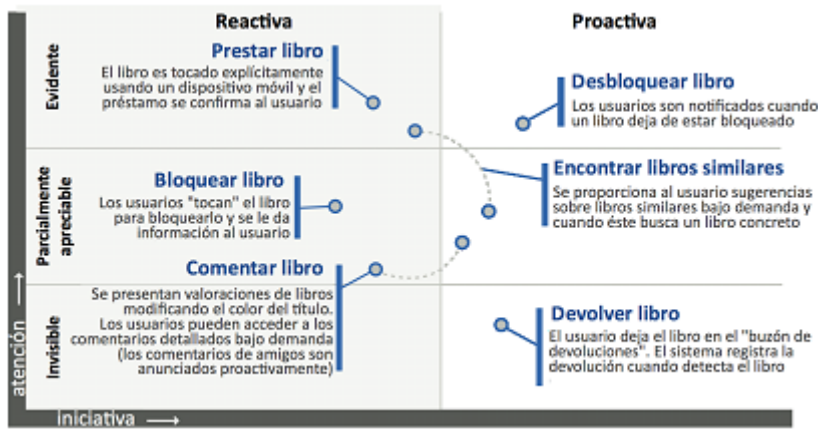


Figura 3. Nivel de intromisión definido para cada tarea de la Biblioteca Inteligente.

reactiva y proactiva. El eje de atención se dividió en tres segmentos, con los siguientes valores: *invisible* (la interacción pasa inadvertida), *parcialmente apreciable* (por norma general, el usuario no la percibirá a menos que haga un esfuerzo), y *evidente* (el usuario es completamente consciente que está llevando a cabo una tarea). Los diseñadores pueden dividir cada eje en tantas partes como requieran para describir el nivel de intromisión del proceso. Esta división será luego utilizada para seleccionar los mecanismos de interacción apropiados para cada elemento que participa en el proceso.

Durante el análisis se definieron los niveles adecuados de intromisión para algunas tareas. En el ejemplo, *Prestar libro* se lleva a cabo de una manera explícita para los usuarios. Éstos inician la interacción (reactiva) y son informados sobre los préstamos (evidente). La tarea *devolver libro* se lleva a cabo de manera totalmente desapercibida para el usuario. Cuando el usuario deja un libro en el buzón de devoluciones, el sistema inicia el proceso de *devolución del libro* sin notificar al usuario. Otras tareas como *encontrar libros similares* se soportan en diferentes niveles de intromisión. En el ejemplo, las *sugerencias* pueden obtenerse por los usuarios explícitamente, o de manera proactiva por el sistema cuando el usuario está cerca de uno de los libros relacionados.

La división definida en el espacio de intromisión permite a los diseñadores clasificar las diferentes técnicas de interacción disponibles. De esta manera, será posible elegir a posteriori la mejor técnica de interacción que encaje con los requisitos capturados. La siguiente sección proporciona más detalles sobre las técnicas de interacción.

4.2. Técnicas de interacción

Los usuarios pueden interactuar con un objeto físico de diferentes maneras. Por ejemplo, pueden acceder a los servicios que están asociados con un objeto bien apuntándolo, to-

cándolo o escaneando los objetos cercanos con su dispositivo móvil. Estos son sólo algunos ejemplos de técnicas de interacción que han sido definidas en la literatura para soportar la interacción entre los usuarios y el entorno que le rodea [15][16][17][18]. Sin embargo, no hay una técnica de interacción universal que sea adecuada para cada situación. Para determinar el nivel de intromisión que debe soportar cada tarea, en este trabajo hacemos uso del marco conceptual de interacción implícita presentado en la **sección 4.1**. Por ejemplo, la técnica de interacción *tocar* puede usarse en las interacciones que se dan en la región *reactiva*, en términos de iniciativa, ya que los usuarios son los que inician la interacción tocando explícitamente el objeto físico con el dispositivo móvil.

Considerando el nivel de intromisión requerido por cada tarea, podemos determinar la técnica de interacción a usar. Por ejemplo, *tocar* parece apropiado para que los usuarios tomen prestados libros, mientras que *escanear* puede usarse para devolver libros de manera

silenciosa. El siguiente paso en el método es especificar cómo encajan estas diferentes interacciones en el proceso de negocio, usando la notación BPMN.

La **figura 4** muestra un diagrama BPMN para el ejemplo de la tarea *prestar libro* de la biblioteca. Los diagramas BPMN se dividen en diferentes líneas, cada una de las cuales representa un participante del proceso (por ejemplo, los *bibliotecarios* o los *socios de la biblioteca*). Cada participante es responsable de llevar a cabo diferentes tareas (rectángulos redondeados), siguiendo un flujo (flechas en el diagrama). Finalmente, existen algunos elementos de control llamados "pasarelas" (diamantes en la figura) que se usan para sincronizar y paralelizar el proceso de negocio. Hemos extendido la notación estándar BPMN [19] para dar soporte a describir la participación de los objetos físicos en los procesos de negocio. La **figura 4** muestra las técnicas de interacción que se usan en cada tarea del proceso de negocio, así como una imagen de la posible implementación de la técnica de interacción usada en el contexto de la biblioteca.

Los diseñadores pueden encontrar en la literatura guías para determinar los pros y contras de cada técnica [16][20]. Por ejemplo, la técnica de interacción de *escanear* puede seleccionarse para devolver libros en el ejemplo ya que permite que estos libros se detecten de manera automática, con una intervención mínima del usuario. Sin embargo, esta técnica depende de la tecnología de identificación que se use para este objeto físico. Por ejemplo, la técnica de *tocar* no se puede usar si el objeto físico sólo se puede identificar por medio de código de barras. La siguiente sección describe el proceso de seleccionar la tecnología de soporte más adecuada dada una técnica de interacción.

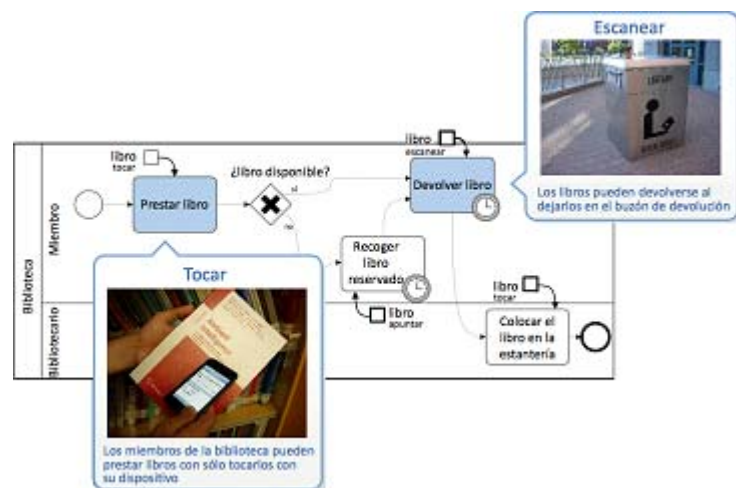


Figura 4. Diferentes interacciones con objetos físicos orquestadas en un modelo de proceso de negocio.

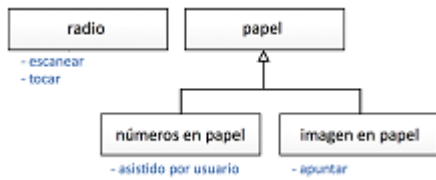


Figura 5. Un ejemplo de diferentes medios y relaciones de especialización.

4.3. Selección de la tecnología

Durante mucho tiempo han existido diferentes tecnologías para unir la separación entre el mundo físico y el virtual [21]. De esta manera, los objetos físicos pueden identificarse de numerosas maneras (por ej. usando una secuencia de barras en papel, o una onda de radio emitida por una etiqueta RFID). La selección de la más adecuada depende de diferentes factores. Por ejemplo, la tecnología RFID no es apropiada para usuarios de una biblioteca si quieren prestar libros de manera autónoma con sus dispositivos móviles ya que la capacidad para leer etiquetas RFID no está presente en la mayoría de dispositivos móviles, en comparación con una cámara. Para tomar este tipo de decisiones es necesario entender el impacto que cada tecnología puede tener en el proceso. Por ejemplo, podemos preguntarnos si existe otra tecnología que pueda ser usada para reemplazar la tecnología RFID y que dé soporte al proceso de negocio como fue descrito (y en el caso de cambios en los requisitos, que se puedan identificar fácilmente).

Definimos el concepto de *medio* para organizar el conocimiento sobre las diferentes tecnologías de identificación. Los medios se definen en este trabajo como los soportes físicos de los identificadores. De esta manera, los diseñadores pueden definir los medios para representar un tipo de tecnologías de una manera abstracta. Por ejemplo, un diseñador puede definir el medio *papel* (ver figura 5) para describir de manera abstracta un conjunto de tecnologías de identificación que tienen requisitos en común: (1) son eficientes en coste, ya que usan identificadores económicos de producir y que requieren dispositivos simples para su captura; y (2) requieren una línea de visión directa, ya que los identificadores deben reconocerse de manera óptica. En este punto del diseño, no estamos interesados en las tecnologías específicas usadas (códigos de barras, marcadores visuales, códigos QR, etc.), pero sí en la categorización más amplia de estas tecnologías con propiedades comunes. En particular, nos interesan las técnicas de interacción que soportan.

El medio es un concepto independiente de tecnología para describir requisitos de identificación. Este concepto puede usarse para guiar el proceso de selección de la tecnología.

Los diseñadores pueden organizar los medios en una jerarquía para capturar de forma más adecuada los aspectos comunes y variables. En este ejemplo, el medio *papel* se especializa en dos medios que se consideran subtipos. Estos subtipos distinguen entre un identificador expresado como números, de otro expresado como una imagen. En el ejemplo de la figura 5, la imagen en papel y los números en papel son medios basados en el papel, pero los mecanismos de interacción que soportan son diferentes. La imagen en papel soporta el mecanismo de *apuntar*, mientras que los números en papel *requieren que los usuarios* identifiquen e introduzcan el identificador asociado (es decir, una interacción mediada por el usuario).

Con este análisis se puede determinar el impacto de la selección de tecnología en el proceso. Por ejemplo, si el RFID (basado en el medio *radio*) es considerado como no apropiado para su uso por los socios de la biblioteca debido a la falta masiva en el mercado de dispositivos móviles con capacidades RFID, podemos explorar las consecuencias de usar otra tecnología de identificación como la basada en el *papel*. En este caso, usar *marcadores visuales* (basados en una imagen sobre el medio papel) puede requerir el uso de técnicas de interacción de *apuntar* a los objetos, en vez de *tocarlos*. La decisión final dependerá del diseñador, pero la propuesta permite trazar el impacto de estas decisiones sobre el proceso de negocio. La figura 6 muestra las tecnologías de identificación seleccionadas para diferentes tareas de la Biblioteca Inteligente (ver parte inferior de cada figura).

Este estudio de los aspectos comunes y la variabilidad de las propiedades de los medios también es útil para dar soporte a la evolución del proceso de negocio, ya que puede ser necesario considerar nuevas tecnologías cuando los procesos de negocio son actualizados. Al clasificar estas nuevas tecnologías dentro de la jerarquía de medios, es posible determinar cuando están realmente proporcionando valor añadido, y cuando no.

4.4. Implantación de la solución

La funcionalidad que soporta un proceso de negocio móvil normalmente se distribuye usando diferentes recursos de computación. Para sustentar la integración de objetos físicos

en las diferentes tareas de un proceso de negocio, se debe organizar la funcionalidad de identificación. Esto implica definir la configuración de los diferentes recursos por el sistema. Usamos el concepto de *unidad de implantación* para encapsular la funcionalidad requerida para soportar las diferentes tareas por medio de un conjunto de tecnologías que se despliegan en un dispositivo concreto.

Para proporcionar una vista abstracta de la configuración de la Biblioteca Inteligente, definimos explícitamente las diferentes unidades de implantación. La figura 6 muestra un diagrama de las unidades de implantación de la Biblioteca Inteligente relacionadas con la tecnología de Auto-ID.

Deben definirse las siguientes propiedades para cada unidad de implantación: **tarea** que soporta, **objeto físico** que está involucrado y **tecnologías** que se usan. En la figura pueden verse las siguientes unidades de implantación: La unidad de implantación *Móvil del Socio* representa un conjunto de componentes software que soportan las tareas de *Prestar libro* y *Recoger libros reservados*, usando tecnologías de códigos QR para ello. Esta unidad de implantación será accesible desde el dispositivo móvil de los socios de la biblioteca. La unidad de implantación *Buzón Devoluciones* es la encargada de detectar automáticamente los *libros devueltos* por medio de tecnología RFID. De esta manera, cada buzón de devoluciones requiere una o varias antenas RFID capaces de detectar su contenido. La unidad de implantación *Móvil del Bibliotecario* se accede desde los dispositivos móviles de los bibliotecarios que deberán estar dotados de antenas RFID para poder dar soporte a la tarea de *devolver libro a la estantería*. Finalmente, la unidad de implantación *Detector Estantería* también da soporte a la tarea *devolver libro en estantería*. En este caso, detecta cuando un libro es ubicado en una estantería incorrecta.

Cuando se definen las diferentes unidades de implantación, es esencial que la tecnología seleccionada sea consistente con los otros aspectos considerados durante la fase de diseño. Esto es, que la tecnología seleccionada debe usar un medio que permita una cierta técnica de interacción para una tarea, y esta técnica soporte un cierto nivel de intrusión para la tarea. Por ejemplo, la unidad de im-



Figura 6. Servicios de Auto-ID involucrados en la Biblioteca Inteligente.

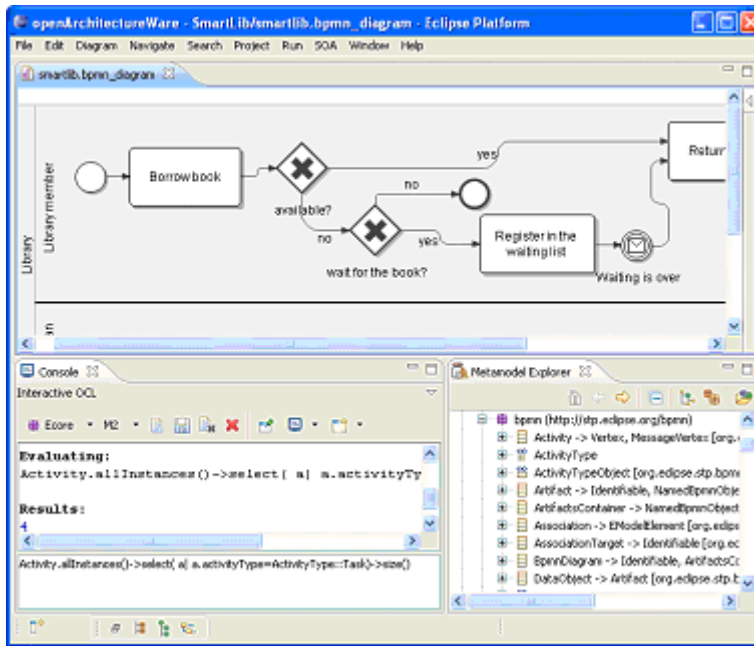


Figura 7. Herramienta basada en Eclipse para editar y consultar modelos de procesos de negocio.

plantación *Móvil del Socio* usa códigos QR que se basan en una imagen sobre papel para dar soporte a la técnica de interacción *apuntar un objeto*, ya que consideramos que esta técnica era adecuada para reemplazar la técnica de *tocar un objeto* (como se discutió en la **sección 4.3**). Este tipo de validaciones pueden ser complicadas de llevar a cabo cuando el proceso de negocio crece. Por esta razón soportaremos la automatización del proceso a través de herramientas software.

5. Herramienta de soporte

Parkour introduce conceptos de diseño para especificar diferentes aspectos para enlazar el mundo físico-virtual en un proceso de negocio. La herramienta que lo soporta proporciona la funcionalidad de modelar y validar procesos de negocio siguiendo el método descrito en este trabajo.

Para especificar los modelos de procesos de negocio usamos como base el metamodelo BPMN definido en el proyecto SOA Tools Platform (STP). Este metamodelo STP define los constructores de modelado para el lenguaje de modelado BPMN, y da soporte muy completo a la especificación de BPMN, incluyendo todas las formas, conexiones y marcadores de BPMN, a excepción de disposiciones gráficas y apariencia de las líneas (*swim lanes*) y la agrupación de artefactos. El proyecto STP también proporciona un editor funcional de diagramas BPMN (ver parte superior de la **figura 7**) que está integrado en las herramientas de modelado basadas en Eclipse.

Además del soporte a la edición, también proporcionamos funcionalidad para verificar

que la descripción del enlace entre el mundo físico-virtual es consistente. Las herramientas de modelado de Eclipse se usan para formalizar estos conceptos introducidos en Parkour y especificar las diferentes restricciones. Por ejemplo, la siguiente expresión comprueba cuando un medio específico soporta una técnica de interacción particular, teniendo en cuenta la jerarquía de medios:

Boolean

```
soportaTecnica (Medio this,
Tecnica Interaccion inter) :
this.tecnicaInteraccion.contains
```

```
(inter) ||
(this.parent != null?
this.parent.soportaTecnica
(inter) : false) ;
```

Al forzar la comprobación de las restricciones, se pueden detectar automáticamente inconsistencias en las especificaciones de procesos de negocio. Esto permite a los diseñadores anticipar la detección de problemas en los procesos de negocio antes de invertir esfuerzo en la implementación e implantación.

De esta manera, los diseñadores pueden estudiar el impacto de quitar, añadir o modificar una tecnología de identificación específica, simplemente modificando el modelo. Esto permite responder preguntas como: *¿Pueden los socios de la biblioteca usar la tecnología RFID para prestar libros usando la técnica de apuntar al objeto?*

La **figura 8** muestra una instancia de modelo Parkour donde el editor ha detectado algunas inconsistencias. Éste verifica las restricciones cada vez que el modelo se guarda (o bajo demanda del usuario). Los errores y avisos detectados se integran en la vista de errores estándar de Eclipse. En el ejemplo, el modelo contiene un error y un aviso. El error se produce porque la unidad de implantación soporta la tarea *prestar libro* por medio de tecnología RFID, mientras que el libro sólo está etiquetado con un identificador basado en papel. Por tanto, la tarea no puede soportarse con la tecnología definida.

6. Conclusiones

En este trabajo hemos presentado un método de diseño para especificar procesos de nego-

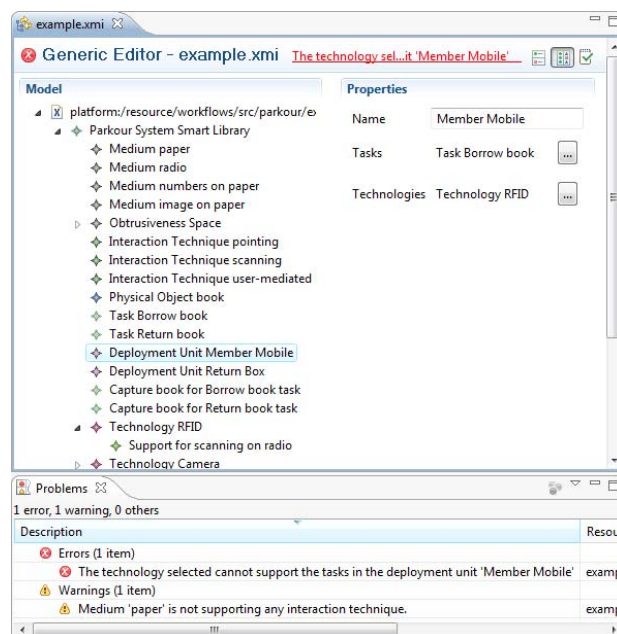


Figura 8. Validación de la restricción de las unidades de implantación.

cio que integran objetos del mundo físico. Se ha mostrado cómo capturar los requisitos para el enlace físico-virtual de una manera gradual, usando técnicas de modelado. El uso de modelos ha sido útil para centralizar el conocimiento sobre el proceso de negocio y organizarlo de una manera sencilla para manipular por los diseñadores (por ej. trabajar con conceptos independientes de tecnología, detectar inconsistencias, etc.).

El método de diseño se basa en técnicas y marcos de trabajo para el modelado de procesos de negocio probados, diseño de interacción implícita y patrones de interacción físicos. La herramienta de soporte a Parkour ha sido útil para anticiparse en la detección de errores en el proceso de negocio descrito. Sin embargo, el uso de herramientas requiere de un conocimiento avanzado en técnicas de modelado.

Este trabajo forma parte de una propuesta integral que incluye un *framework* llamado Presto y generadores de código para dar soporte completo al método de desarrollo presentado en la **figura 1**. Presto define una arquitectura que encaja con las necesidades específicas de los procesos de negocio físicos. Esta arquitectura ha sido definida con independencia de la tecnología. Así, los componentes de esta arquitectura pueden mapearse a diferentes plataformas tecnológicas. Nuestra propuesta da soporte para obtener automáticamente un diseño arquitectónico para una especificación de proceso de negocio dada. La generación de código permite producir los componentes de implementación requeridos por la infraestructura tecnológica de la plataforma destino a partir de los modelos especificados. En particular, se ha definido el soporte para la plataforma Android que evita a los desarrolladores tratar con los componentes específicos de Android y les permite centrarse en implementar la lógica de negocio requerida en Java.

Agradecimientos

Este trabajo se ha desarrollado con la financiación del MICINN en el ámbito de proyecto EVERYWARE con referencia TIN2010-18011.

Referencias

- [1] Neil Gershenfeld, Raffi Krikorian. The Internet of Things. *Scientific American*, 291(4):pp. 46-51, 2004.
- [2] Martin Strassner, Thomas Schoch. Today's Impact of Ubiquitous Computing on Business Processes. En Friedemann Mattern, Mahmoud Nagshineh editors, *Short Paper Proc. of International Conference on Pervasive Computing*, pp. 62-74, 2002.
- [3] Jean Marie Favre. Foundations of Model (Driven) (Reverse) Engineering – Episode I: Stories of the Fidus Papyrus and of the Solarus. En Jean Bézin, Reiko Heckel editors, *Post-Proceedings of Dagstuhl Seminar on Model Driven Reverse Engineering*, Germany, 2004.
- [4] Marc Langheinrich, Vlad Coroama, Juergen Bohn, Michael Rohs. *As we may live - Real-world implications of ubiquitous computing*. Technical report, Swiss Federal Institute of Technology, 2002.
- [5] Elgar Fleisch. Business perspectives on Ubiquitous Computing. Technical report, M-Lab Working, Switzerland, 2001.
- [6] Uwe Sandner, Jan Marco Leimeister, Helmut Krcmar. Business Potentials of Ubiquitous Computing. En Eva-Maria Kern et al. editors, *Managing Development and Application of Digital Technologies*, pp. 277-292. Springer, 2006.
- [7] Andrew E. Fano, Anatole Gershman. The future of business services in the age of ubiquitous computing. *Commun. ACM*, 45(12):pp. 83-87, 2002.
- [8] Elgar Fleisch, Christian Tellkamp. The Challenge of Identifying value-creating Ubiquitous Computing applications. *Proc. of the Workshop on Ubiquitous Commerce, Ubicomp*, Seattle, USA, 2003.
- [9] Sanjay E. Sarma. Integrating RFID. *ACM Queue*, 2(7):pp. 50-57, 2004.
- [10] George Roussos, Juha Tuominen, Leda Koukara, Olli Seppala, Panos Kourouthanasis, George M. Giaglis, Jeroen Frissaer. A Case Study in Pervasive Retail. En Marisa-S. Viveros et al. editors, *Proc. of the Workshop Mobile Commerce*, pp. 90-94. ACM, 2002.
- [11] Jeff Kramer. Is abstraction the key to computing? *Commun. ACM*, 50(4):pp. 36-42, 2007.
- [12] Douglas C Schmidt. Model Driven Engineering. *Computer. Guest Editor's Introduction.*, 39(2):pp. 25-31, 2006.
- [13] Pau Giner, Carlos Cetina, Joan Fons, Vicente Pelechano. Developing mobile workflow support in the internet of things. *IEEE Pervasive Computing*, 9(2):pp. 18-26, 2010.
- [14] Wendy Ju, Larry Leifer. The Design of Implicit Interactions: Making Interactive Systems Less Obnoxious. *Design Issues*, 24(3):pp. 72-84, 2008.
- [15] Liviu Iftode, Cristian Borcea, Nishkam Ravi, Porlin Kang, Peng Zhou. Smart Phone: An Embedded System for Universal Interactions. *Proc. of Future Trends of Distributed Computing Systems*, pp. 88-94. IEEE Computer Society, 2004.
- [16] Enrico Rukzio, Karin Leichtenstern, Victor Callaghan, Paul Holleis, Albrecht Schmidt, Jeannette Shiaw-Yuan Chin. An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. En Paul Dourish, Adrian Friday editors, *Proc. of International Conference on Ubiquitous Computing, UbiComp*, volume 4206 of *Lecture Notes in*

- Computer Science*, pp. 87-104. Springer, 2006.
- [17] Gregor Broll, Markus Haarländer, Massimo Paolucci, Matthias Wagner, Enrico Rukzio, Albrecht Schmidt. Collect&Drop: A Technique for Multi-Tag Interaction with Real World Objects and Information. En Emile H. L. Aarts et al. editors, *Proc. of the European Conference on Ambient Intelligence, Aml*, volume 5355 of *Lecture Notes in Computer Science*, pp. 175-191. Springer, 2008.
- [18] Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan, Jan Borchers. BYOD: Bring Your Own Device. *Proc. of the Workshop on Ubiquitous Display Environments, Ubicomp*, Tokyo, Japan, 2004.
- [19] Pau Giner, Victoria Torres, Vicente Pelechano. Modeling Mobile Business Processes for the Internet of Things. *Proc. of the Workshop on Web Services and SOA, JSWEB*, Madrid, Spain, 2009.
- [20] Enrico Rukzio, Gregor Broll, Karin Leichtenstern, Albrecht Schmidt. Mobile Interaction with the Real World: An Evaluation and Comparison of Physical Mobile Interaction Techniques. En Bernd Schiele et al. editors, *Proc. of the European Conference on Ambient Intelligence, Aml*, volume 4794 of *Lecture Notes in Computer Science*, pp. 1-18. Springer, 2007.
- [21] Roy Want, Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison. Bridging Physical and Virtual Worlds with Electronic Tags. *Proc. of the SIGCHI Conference on Human Factors in Computer Systems, CHI*, pp. 370-377, 1999.