

De los Procesos del Negocio a los Casos de Uso¹

Jesús García Molina, M. José Ortín, Begoña Moros, Joaquín Nicolás, Ambrosio Toval

Grupo de Investigación de Ingeniería del Software²
Departamento de Informática y Sistemas
Facultad de Informática. Universidad de Murcia
C.P. 30.071 Campus de Espinardo, Murcia, Spain
{jmolina, mjortin, bmoros, jnr, atoval}@um.es

Modelo Conceptual

Diagrama de Actividad

Casos de Uso

Resumen. En este trabajo se presenta una estrategia para obtener de modo sistemático el modelo de casos de uso y el modelo conceptual, a partir del modelado del negocio basado en diagramas de actividades UML. Después de determinar los procesos del negocio de la organización bajo estudio, y de describir sus flujos de trabajo mediante diagramas de actividad, los casos de uso son identificados y estructurados a partir de las actividades de cada proceso, mientras que los conceptos que aparecen en el modelo conceptual se obtienen a partir de los datos que fluyen entre las actividades. Además, las reglas del negocio son identificadas e incluidas en un glosario, como parte de la especificación de datos y actividades. Un aspecto destacable de nuestra propuesta es el hecho de que el modelado conceptual y el de casos de uso se realiza en paralelo, haciendo más fácil la identificación y especificación de casos de uso adecuados. Tanto el modelado de casos de uso como el modelado conceptual forman parte de la fase de análisis de requisitos de un modelo de proceso completo en cuya definición estamos trabajando. Este proceso está siendo experimentando en un organismo de tamaño medio de la Administración Autonómica.

1 Introducción

Desde que UML [1] fue adoptado por el OMG como el lenguaje estándar para el modelado, se ha definido un buen número de modelos de proceso para el desarrollo de aplicaciones orientadas a objetos (OO), que utilizan este lenguaje como medio de expresión de los diferentes modelos que se crean durante el desarrollo. Estas propuestas suelen estar *dirigidas por los casos de uso*, de manera que éstos se emplean para definir los requisitos funcionales del sistema, y todas las etapas del proceso (planificación de las iteraciones, análisis, diseño y pruebas) se articulan en torno a los casos de uso identificados.

Actualmente, en muchas discusiones sobre casos de uso. Se coincide en señalar que con frecuencia son mal interpretados, y que no hay guías precisas para resolver los aspectos que tienen que ver con su organización. En este sentido, se han publicado diferentes propuestas (por ejemplo [3, 7, 8]) en las que se discuten cuestiones tales

¹ Parcialmente subvencionado por la CICYT, Ministerio de Educación y Ministerio de Industria. Proyecto de investigación MENHIR-OM TIC97-0593-C05-02.

² Miembro de RENOIR (*European Requirements Engineering Network of Excellence*).

como la granularidad de los casos de uso, el nivel de detalle en que deben describirse, o la conveniencia de crear una jerarquía de casos de uso.

Inspirados en la *arquitectura de tres modelos de OOram* [13] y en el método *IDEA* [2], estamos definiendo un proceso basado en UML orientado a sistemas de información de gestión. Este proceso incluye una fase de modelado del negocio, que describe los procesos del negocio de la organización bajo estudio de manera que se puedan construir, de forma sencilla y directa, versiones iniciales de los modelos conceptual y de casos de uso. Cada proceso del negocio se describe haciendo uso de un diagrama de actividades UML con calles (swimlanes). Posteriormente, se identifican los casos de uso del sistema a partir de las actividades y los *conceptos* (clases del dominio) a partir de los datos (objetos de información que fluyen entre las actividades).

En este trabajo describimos nuestra propuesta para realizar el modelado del negocio y su conexión con el análisis de requisitos (modelos conceptual y de casos de uso). Esta propuesta ha sido experimentada en el marco de un proyecto cuyo objetivo ha sido proporcionar un modelo de proceso, basado en requisitos, para el desarrollo de sistemas de información de gestión con uso intensivo de datos [10]. El ámbito de este trabajo ha sido la DGSIC (*Dirección General de Servicios de Información y de las Comunicaciones*) de la CARM (*Comunidad Autónoma de la Región de Murcia*).

Este trabajo está estructurado de la siguiente manera: en el apartado 2 comentamos someramente la **problemática** asociada a la utilización del concepto de caso de uso, y ofrecemos una visión general de nuestra propuesta; en el apartado 3 presentamos la manera de abordar el **modelado del negocio**; en el apartado 4 mostramos cómo realizar la **transición** desde el modelo del negocio a los modelos de casos de uso y conceptual; finalmente, en la sección 5 exponemos nuestras conclusiones.

- 2. Problemas con CU
- 3. Modelado Negocio
- 4. Transición MN a CU

2 Motivación

2.1 Problemas en la Utilización de los Casos de Uso

Actualmente, la mayor parte de los modelos de proceso propuestos para UML se definen como *dirigidos por los casos de uso*. Un caso de uso puede ser definido como una secuencia de acciones, incluyendo variaciones, que el sistema puede ejecutar y que produce un resultado observable de valor para un actor que interactúa con el sistema [1]. Aunque el éxito de los casos de uso se suele justificar con el hecho de que constituyen una técnica simple e intuitiva, algunos autores (ver por ejemplo [3, 7, 8]) señalan las dificultades que entraña la obtención y la especificación de casos de uso verdaderamente útiles, y la falta de consenso sobre cómo organizarlos y manejarlos. Estas son las razones que nos llevan a pensar que es necesario establecer un conjunto de guías para la identificación, descripción y organización de los casos de uso.

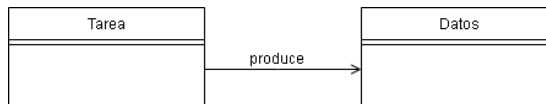
Algunas discusiones interesantes acerca del manejo de casos de uso son las proporcionadas por T. Korson y A. Cockburn. Korson [7] defiende que los requisitos (y por tanto los casos de uso) han de ser organizados jerárquicamente, y establece que i) cada nivel de casos de uso no debe añadir nuevos requisitos, sino refinar los del nivel superior, y ii) la jerarquía de casos de uso no debe ser el resultado de una **descomposición funcional**, y ha de ser desarrollada de manera iterativa e incremental.

Por otro lado, Cockburn [3] utiliza el concepto de objetivo (*goal*) para organizar jerárquicamente los casos de uso. Distingue básicamente entre *objetivos estratégicos*

Actividades ⇒ Casos de Uso
Datos ⇒ Entidades Conceptuales

obtienen de los datos que fluyen entre tales actividades. Además, se identifican las reglas del negocio y se incluyen en un glosario como parte de la especificación de los datos y las actividades. Un aspecto notable de nuestra propuesta es que el modelado de casos de uso y el modelado conceptual se realizan al mismo tiempo, haciendo más fácil, por tanto, la identificación y especificación de los casos de uso adecuados.

3 Modelado del Negocio



Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de *procesos de negocio*. Cada uno de ellos se caracteriza por una colección de *datos* que son *producidos* y manipulados mediante un conjunto de *tareas*, en las que ciertos *agentes* (por ejemplo, trabajadores o departamentos) participan de acuerdo a un *flujo de trabajo* determinado. Además, estos procesos se hallan sujetos a un conjunto de *reglas de negocio*, que determinan las políticas y la estructura de la información de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior. De acuerdo con el concepto de objetivo estratégico de Cockburn [3], capturamos los procesos del negocio a partir de los objetivos principales de la empresa. En primer lugar, consideramos los objetivos estratégicos de la organización. Teniendo en cuenta que estos objetivos van a ser muy complejos y de un nivel de abstracción muy alto, serán descompuestos en un conjunto de subobjetivos más concretos, que deberán cumplirse para conseguir el objetivo estratégico. Estos subobjetivos pueden a su vez ser descompuestos en otros, de manera que se defina una jerarquía de objetivos. En nuestro estudio, hemos experimentado que dos o tres niveles de descomposición son suficientes. Para cada uno de estos subobjetivos de segundo nivel definimos un proceso de negocio que deberá dar soporte a dicho subobjetivo. Representamos cada proceso del negocio como un *caso de uso del negocio*, que inicialmente será descrito de forma textual.

En el resto del trabajo, ilustramos el proceso mediante el ejemplo de una compañía que fabrica productos bajo demanda (siguiendo un esquema *just in time*). Los objetivos estratégicos de dicha compañía podrían incluir *Satisfacer un pedido de un cliente*, *Incrementar en un 25% las ventas*, o *Disminuir el tiempo de fabricación en un 15%*. El objetivo *Satisfacer un pedido de un cliente* puede ser dividido en subobjetivos tales como: *Registrar Pedido de Cliente*, *Fabricar Producto Pedido*, *Gestionar Almacén* y *Realizar Pedidos a Proveedores*. Éstos serán los objetivos que utilizaremos para definir nuestros procesos del negocio.

3.1 Identificación de Roles del Entorno del Negocio

Una vez se han identificado los procesos de negocio, es preciso encontrar los agentes involucrados en su realización. Cada uno de estos agentes o actores del negocio desempeña cierto papel (*juega un rol*) cuando colabora con otros para llevar a cabo las

SEÑALADOR

Primer paso:
Definir los
procesos de
negocio

Límites

actividades que conforman dicho caso de uso del negocio. De hecho, identificaremos los roles que son jugados por agentes de la propia empresa (que incluyen trabajadores, departamentos y dispositivos físicos) o agentes externos (como clientes u otros sistemas). Por el momento nos centraremos en este último tipo de roles, con los que la organización interactúa para llevar a cabo sus procesos de negocio. **En nuestro ejemplo tenemos los roles *Cliente* y *Proveedor*, claramente externos al sistema.**

ROLES:
-Agentes propios
-Agentes externos

Para tener una visión general de los diferentes procesos de negocio de la organización, puede construirse un *diagrama de casos de uso del negocio*, en el cual aparece cada proceso del negocio como un caso de uso. Este diagrama permite mostrar los límites y el entorno de la organización bajo estudio. Sólo se mostrarán en este diagrama los actores del negocio correspondientes a los roles externos al sistema, de forma que los procesos de negocio en los que sólo tomen parte roles internos a la organización no estarán conectados a ningún actor. En la Fig. 2 se muestra el diagrama de casos de uso del negocio para nuestro ejemplo; es un diagrama de casos de uso UML formado por casos de uso del negocio y actores. En el diagrama se muestra además que el agente *Cliente* arranca la realización del caso de uso relacionado, mientras que *Proveedor* simplemente participa en el caso de uso asociado.

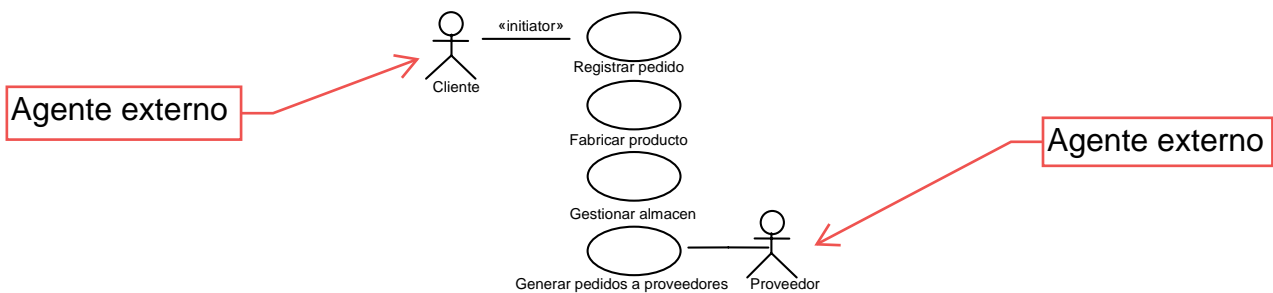


Fig. 2. Diagrama de casos de uso del negocio para el sistema de producción *just in time*

3.2 Descripción de los Casos de Uso del Negocio

PASO 2 ⇒

El siguiente paso dentro del modelado del negocio es introducirse en cada uno de los casos de uso del negocio identificados para describirlos en detalle. Nos centraremos en uno de los casos de uso del negocio de nuestro ejemplo, *Registrar Pedido*, cuya descripción se muestra en la Fig. 3. Esta descripción puede ser validada fácilmente por los usuarios.

A continuación, hemos de determinar los agentes internos que juegan un rol en cada caso de uso del negocio. Hasta el momento hemos identificado los roles que pertenecen al entorno de la organización. Ahora es necesario estudiar la descripción de cada caso de uso del negocio, y observar el conjunto completo de roles involucrados, tanto externos como internos a la organización. Los roles del caso del uso del negocio *Registrar pedido* son *Cliente*, *Comercial*, *Jefe_Técnico*, y *Jefe_Producción* (siendo los tres últimos internos al sistema).

1. El cliente envía una orden de pedido, que debe incluir la fecha de solicitud, datos del cliente y productos solicitados. Es posible que sea un empleado del departamento comercial quien introduzca el pedido, a petición de un cliente que realizó su pedido por teléfono o lo envió por fax o correo ordinario al depto. comercial de la empresa.
2. El empleado revisa el pedido (completándolo, si es necesario), y comienza su procesamiento enviándolo al jefe técnico, que está encargado de su análisis.
3. El jefe técnico analiza la viabilidad de cada producto del pedido por separado:
 - Si el producto pedido está en el catálogo, su fabricación es aceptada.
 - En caso contrario, es considerado un *producto especial*, y el jefe técnico estudia su producción:
 - Si es viable, la fabricación del producto especial es aceptada;
 - Si no es viable, el producto especial no será fabricado.
4. Una vez estudiado el pedido completo, el jefe técnico...
 - Informa al departamento comercial de la aceptación o rechazo de cada producto pedido;
 - Si todos los productos de un pedido han sido aceptados, se crea una orden de trabajo para cada producto, a partir de una plantilla de fabricación (la estándar si el producto estaba catalogado, o una nueva, específicamente diseñada para el producto, si éste no estaba en el catálogo). Cada orden de trabajo es enviada al jefe de producción, y queda pendiente de su lanzamiento.
5. El comercial comunica al cliente el resultado final del análisis de su pedido.

Fig. 3. Descripción del caso de uso del negocio *Registrar pedido*

El aspecto estructural de la colaboración entre los roles para llevar a cabo un caso de uso del negocio, puede ser representado en un *diagrama de roles*, en el que cada rol (una clase UML estereotipada) aparece asociado con los roles con los que puede colaborar (ver Fig. 4). Por tanto, este diagrama permite expresar el conocimiento que unos roles tienen de otros, así como las características (como la multiplicidad) de cada relación entre roles. Además, este diagrama permite también mostrar las características de los roles identificados, tales como sus atributos y responsabilidades. Ortín y García Molina [11] discuten con más detalle el modelado de roles con UML.

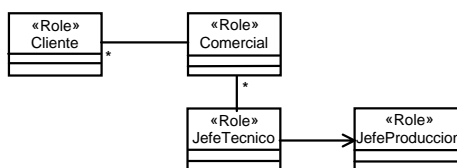


Fig. 4. Diagrama de roles para el caso de uso del negocio *Registrar Pedido*

Después crearemos escenarios para mostrar el aspecto de comportamiento de la colaboración. Para ello utilizaremos *diagramas de secuencia* UML (ver Fig. 5), en los que los objetos denotan las instancias de los roles que intervienen en la interacción.

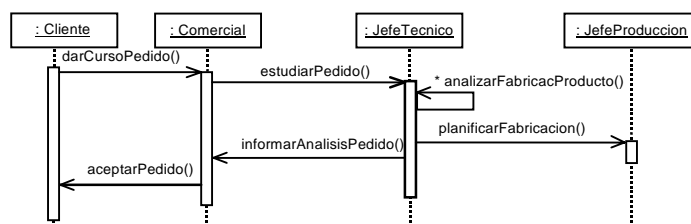


Fig. 5. Diagrama de secuencia para el caso de uso del negocio *Registrar Pedido*

En cada proceso podemos distinguir entre el flujo básico o normal de la interacción (en nuestro ejemplo, *solicitud de un pedido que es aceptado*) y los posibles flujos

alternativos (por ejemplo, *rechazo o cancelación de un pedido*). Para mejorar la legibilidad, es conveniente asociar varios escenarios a un mismo caso de uso del negocio, en lugar de mostrar en una única secuencia todas las posibilidades.

En la arquitectura de tres modelos de OOram [13] se incluye un modelo del negocio representado mediante una *vista proceso* basada en el estándar IDEF0 [5], en la cual se muestra el flujo de trabajo a realizar para conseguir cierto objetivo de la organización, indicando qué roles realizan cada actividad y cuáles son los datos requeridos y producidos por cada actividad. Creemos que estos diagramas son muy útiles para modelar casos de uso del negocio, dado que son muy sencillos y expresivos, facilitando así la comunicación con los usuarios. Estos diagramas pueden adaptarse a UML utilizando diagramas de actividades con calles (*swimlanes*). De esta manera, para mostrar de forma más detallada el flujo de trabajo que realiza cada proceso de negocio crearemos diagramas de este tipo, que llamaremos *diagramas de proceso*.

La Fig. 6 muestra el diagrama de proceso que incluye el escenario de la Fig. 4. Existe una calle por cada rol participante en el escenario, que incluye las actividades que realiza dicho rol. El diagrama también muestra la información que necesita y produce cada actividad, y la sincronización requerida entre las diferentes actividades. Los datos aparecen como objetos que fluyen entre las actividades y pueden tener un estado. Por ejemplo, la actividad *Cursar pedido* recibe un pedido propuesto e inicia su revisión (ver Fig. 6). Nos referimos a estos objetos como *objetos de información*.

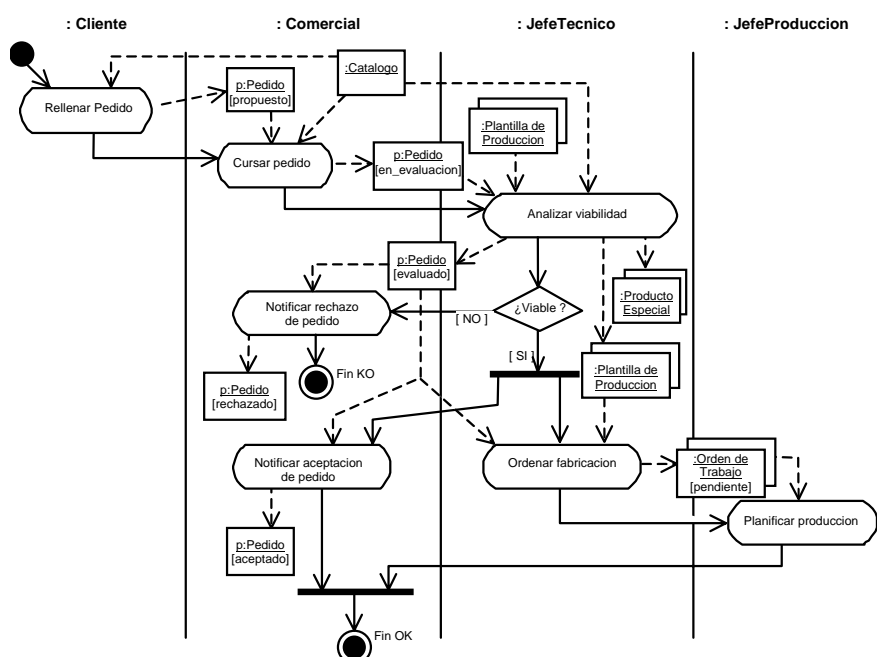


Fig. 6. Diagrama de proceso para el caso de uso del negocio *Registrar Pedido*

Durante la descripción de un proceso de negocio mediante un diagrama de proceso, es posible encontrar una actividad cuya complejidad sea tal que sea necesario describirla mediante otro diagrama de proceso adicional, por no complicar en exceso el

diagrama en cuestión. Por tanto, este nuevo diagrama de proceso describirá un sub-objetivo en relación al objetivo ligado al proceso de negocio original. De este modo los procesos de negocio se organizan jerárquicamente. También es posible mostrar en diferentes diagramas de proceso el flujo normal y los flujos alternativos.

3.3 Especificación de Reglas del Negocio

En una organización, tanto los procesos como los datos que estos manejan, están restringidos por las reglas del negocio. Con el fin de tener en cuenta todos los tipos de reglas que aparecen en la especificación de requisitos, hemos utilizado la clasificación descrita por James Odell [9], que distingue entre *reglas de restricción* (*reglas de estímulo-respuesta*, *reglas de restricción de operación* y *reglas estructurales*) y *reglas de derivación*. De acuerdo con esta clasificación, recogemos de manera explícita cada tipo de regla en el modelo del negocio mediante la especificación de las actividades y objetos de información que aparecen en los diagramas de proceso. Estas especificaciones se reúnen en un glosario. La Fig. 7 muestra la especificación del objeto de información *Pedido* y de las actividades *Ordenar fabricación* y *Notificar aceptación de pedido*.

<p>... Objeto de Información: <u>Pedido</u> Atributos Código de pedido Fecha de solicitud Fecha de creación Fecha máxima de entrega Conjunto de {Producto} Cliente Importe total Estado actual Restricciones <ul style="list-style-type: none"> - El código de pedido identificará unívocamente el pedido, y será asignado automáticamente por el sistema. - Las fechas de solicitud y de creación serán previas a la fecha máxima de entrega. - Un pedido contendrá al menos un producto; no existe límite máximo de productos. - Un pedido siempre será solicitado por un y solamente un cliente - El importe total del pedido será calculado a partir del precio y unidades pedidas de cada producto incluido. ... Clase del Dominio: -pendiente de especificar- ...</p>	<p>... Actividad: <u>Ordenar fabricación</u> Origen: Analizar viabilidad Agente: Jefe Técnico Precondiciones: <ul style="list-style-type: none"> - La fabricación de todo producto en el pedido es viable - Existe una plantilla de fabricación para cada uno de dichos productos. Postcondiciones: <ul style="list-style-type: none"> - Ha sido creada una orden de trabajo para cada producto solicitado; - El estado de cada orden de trabajo es pendiente. - Cada orden de trabajo ha sido enviada al jefe de producción para su planificación. Caso de Uso del sistema: -pendiente de especificar- Actividad: <u>Notificar aceptación de pedido</u> Origen: Analizar viabilidad Agente: Comercial Precondiciones: <ul style="list-style-type: none"> - La fabricación de todos sus productos es viable. Postcondiciones: <ul style="list-style-type: none"> - Se ha comunicado al cliente la aceptación de su pedido. - El estado del pedido es aceptado. Caso de Uso del Sistema: -pendiente. de especificar- ...</p>
--	--

Fig. 7. Extracto del glosario: objetos de información y actividades

Cada objeto de información se describirá mediante un conjunto de atributos y sus restricciones de integridad (si las tuviera); por tanto, establecemos explícitamente las reglas estructurales y de derivación. Por otro lado, la especificación de la semántica de cada actividad contendrá: *origen* (actividades que la preceden), *agente* (responsable de llevar a cabo la actividad), y *pre* y *post-condiciones* (que establecen qué tiene que cumplirse antes y después de la actividad). Esta última parte se corresponde con las reglas de operación, mientras que las reglas de estímulo-respuesta quedan refleja-

das mediante el *origen*, donde se expresa el orden entre las actividades. El glosario tendrá una estructura de hipertexto (referencias-cruzadas) con el objeto de mantener las relaciones de trazabilidad entre los procesos del negocio y las clases y los casos de uso que especifican la funcionalidad del sistema.

4 Análisis de Requisitos: Modelos Conceptual y de Casos de Uso Iniciales

A partir del modelo del negocio descrito en la sección anterior, es posible obtener de manera sistemática y directa, tanto la colección inicial de casos de uso del sistema como el modelo conceptual preliminar. A continuación vamos a describir de manera separada cómo obtener cada modelo.

Los requisitos elicitados y especificados en esta fase serán incluidos en un documento de especificación de requisitos del software (ERS). Recomendamos el uso de una plantilla de ERS estándar, como la IEEE 830-1998.

4.1 Transición al Modelo Inicial de Casos de Uso del Sistema

Según nuestra experiencia, las actividades del diagrama de proceso tienen el nivel de granularidad adecuado para ser asociadas a un caso de uso del sistema. De esta manera, crearemos un caso de uso del sistema por cada actividad del diagrama de proceso que deba ser soportada por el sistema software. Por tanto, el rol que lleva a cabo la actividad será el actor principal del caso de uso. Nótese que, de acuerdo con la definición de caso de uso, no todas las actividades del diagrama de proceso serán consideradas como casos de uso, sino solamente aquellas que sean de valor para algún actor.

Por ejemplo, supongamos que el rol *Cliente* no rellenara él mismo el pedido (mediante un formulario web, por ejemplo), sino que comunicara todos los datos por fax, mediante una llamada telefónica, o por cualquier otro medio, como resultado de la actividad *Rellenar pedido* (ver Fig. 6). Como esta actividad se llevaría a cabo fuera del sistema software, no aparecerían en el diagrama de casos de uso del sistema ni el rol *Cliente* (puesto que no interactúa con el sistema software) ni la actividad *Rellenar pedido*. En la Fig. 8 se muestra el diagrama de casos de uso del sistema para el proceso del negocio *Registrar Pedido*, correspondiente al diagrama de proceso de la Fig. 6, considerando que todas las actividades serán soportadas por el sistema software. El diagrama contiene los casos de uso más importante desde el punto de vista de la arquitectura del sistema.

Debemos señalar que algunos casos de uso no se obtendrán directamente a partir de los diagramas de proceso. Estos nuevos casos de uso se detectarían al describir los casos de uso identificados y adquirir un mayor conocimiento sobre los requisitos que deben ser soportados, y representarían funciones que debe llevar a cabo el sistema para lograr algún objetivo asociado con algún caso de uso ya existente. En nuestro ejemplo, para *Analizar viabilidad* es necesario buscar en el catálogo de productos si un producto solicitado existe y, por tanto, este catálogo debe mantenerse actualizado. En consecuencia, añadimos el caso de uso *Mantener Productos del Catálogo*. Otro ejemplo de un nuevo caso de uso sería *Mantener Plantillas de Fabricación*.

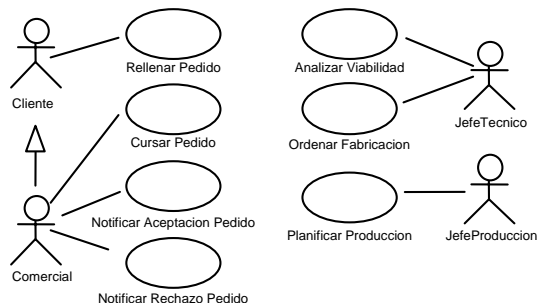


Fig. 8. Diagrama inicial de casos de uso del sistema

Los casos de uso se pueden organizar en varios niveles (recomendamos dos o tres como máximo) de acuerdo con la descomposición jerárquica propuesta en el modelo del negocio.

Cada caso de uso se describirá mediante una plantilla que puede rellenarse a partir de la especificación de la actividad asociada, que se encuentra recogida en el glosario como ya hemos visto. Hemos elegido la plantilla propuesta por Coleman [4] porque combina simplicidad y completitud, como se muestra en la Fig. 9.

Una vez descrito el caso de uso, se conectará a la especificación de la actividad asociada en el glosario, con el objeto de mantener la trazabilidad entre los casos de uso del negocio y del sistema.

Caso de Uso	Ordenar Fabricación
Descripción	Se crearán órdenes de trabajo para cada producto solicitado en el pedido, y serán enviadas al jefe de producción para su planificación.
Actores	Jefe técnico
Asunciones	<ul style="list-style-type: none"> - Es viable la fabricación de cada producto solicitado en el pedido. - Existe una plantilla de fabricación para cada producto solicitado.
Pasos	1 REPETIR <ul style="list-style-type: none"> 1.1 Obtener un producto del pedido. 1.2 Buscar la plantilla de fabricación asociada al producto. 1.3 Crear la orden de trabajo. 1.4 Almacenar la orden de trabajo con el estado <i>pendiente</i>.
Variaciones	--
Req. No Funcionales	--
Cuestiones	--

Fig. 9. Descripción del caso de uso del sistema *Ordenar Fabricación*

También podrían encontrarse relaciones entre los casos de uso, tales como *include*, si se detectan aspectos comunes entre varios casos de uso, y *extend*, para expresar caminos opcionales o alternativos en un caso de uso. No obstante, estamos de acuerdo con las recomendaciones ampliamente extendidas de no abusar de estas relaciones y no mostrarlas en los diagramas de casos de uso.

Para completar esta fase debemos establecer los requisitos no funcionales. Cuando estén asociados a un caso de uso, podrán especificarse en la plantilla de caso de uso propuesta. Los requisitos no funcionales globales se recogerán en el apartado correspondiente de la plantilla de ERS elegida.

4.2 Obtención del Modelo Conceptual Inicial

Los objetos de información que fluyen entre las actividades de un caso de uso del negocio representan datos del dominio, por lo que suponen una buena base para crear el modelo conceptual inicial. Este modelo incluirá los conceptos y sus relaciones y se describirá mediante un diagrama de clases UML, en el que los conceptos se representan mediante clases (clases del dominio).

Así, cada objeto de información del diagrama de proceso se convertirá ahora en un concepto (y en la etapa de diseño dará lugar a una clase si el sistema software debe dar soporte a dicho concepto). A partir de la especificación de un objeto de información obtendremos la definición del concepto asociado, es decir, sus atributos, relaciones con otras clases y restricciones. Por ejemplo, a partir de la especificación de *Pedido* mostrada en la Fig. 7, podríamos obtener: i) los atributos *codigo*, *fechaSolicitud*, *fechaCreacion*, *fechaMaxEntrega*, *importeTotal*, *estadoActual*; ii) las asociaciones *Cliente-Pedido* y *Pedido-Producto*, y iii) restricciones que podrían ser expresadas textualmente o bien mediante OCL (*Object Constraint Language*), como $\{fechaMaxEntrega > fechaCreacion\}$. Nótese además que cuando un modelo conceptual evoluciona hacia un diagrama de clases, las responsabilidades se pueden obtener a partir de ciertas restricciones ya especificadas en el glosario. Por ejemplo, la clase *Pedido* podría tener responsabilidades como *obtenerProductos*, *calcularFechaMaxEntrega*, *calcularImporteTotal* o *cambiarEstado*.

De igual forma que conectábamos en el glosario las actividades con los casos de uso del sistema, vincularemos cada objeto de información con la clase del dominio que lo representa en el sistema. La Fig. 10 muestra el diagrama de clases que describe el primer modelo conceptual de nuestro ejemplo.

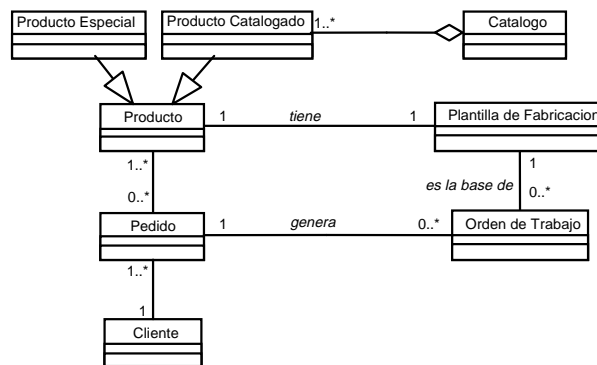


Fig. 10. Modelo conceptual inicial para el caso de uso del negocio *Registrar Pedido*

En esta etapa del desarrollo, merece la pena detenerse en la identificación de los conceptos y no tanto en las relaciones entre ellos. Deberíamos concentrarnos en las asociaciones del tipo *debe-conocer*. Por ejemplo, a partir del glosario podemos establecer que un pedido *debe conocer* al cliente que lo realiza y los productos que lo componen (ver Fig. 7). De esta forma, alguno de los roles identificados en el modelo

del negocio, y por tanto especificado en el modelo de roles, podría ser incluido como una clase en el modelo conceptual. Es el caso de la clase *Cliente* en nuestro ejemplo.

A partir del modelo del negocio, es posible identificar también qué clases tienen un comportamiento que depende de un conjunto no trivial de estados alcanzables. En estos casos, sería interesante definir una máquina de estados mediante un *diagrama statechart* UML. Estas clases se detectan con facilidad en los diagramas de proceso, puesto que se corresponden con objetos de información etiquetados con varios estados diferentes. En nuestro ejemplo, *Pedido* sería candidato para construir una máquina de estados que mostrase los estados de un pedido (*propuesto*, *en_evaluación*, *evaluado*, *aceptado* y *rechazado*) y los eventos que producen los cambios entre estados.

5 Conclusiones

Este trabajo presenta una estrategia para abordar el modelado del negocio y el análisis de requisitos, en la que los casos de uso y el modelo conceptual se obtienen de forma sencilla, a partir del modelo del negocio basado en el uso de diagramas de actividades UML.

Con las guías proporcionadas, el modelador dispone de un modo sistemático de identificar y organizar casos de uso, y de identificar y definir las clases del modelo conceptual. Los procesos de negocio de la organización se identifican partiendo de los objetivos propuestos por Cockburn [3], y se describen mediante flujos de actividades que se representan mediante diagramas de actividades UML. De este modo, los casos de uso del sistema se obtienen a partir de las actividades de los procesos del negocio y se organizan jerárquicamente, de acuerdo con lo indicado por Korson [7].

Las clases del modelo conceptual se obtienen a partir de los objetos de información que fluyen entre las actividades. Nos gustaría subrayar, como una característica importante de nuestro enfoque, que el modelado de los casos de uso del sistema y el modelado conceptual se realizan en paralelo, de acuerdo con Korson [8], quien establece que esto es crucial para obtener casos de uso correctos, puesto que es necesario entender bien el dominio para poder escribir casos de uso que sean realmente útiles.

A la vez que se realiza el modelado del negocio y de los requisitos, la especificación de las actividades y de los casos de uso asociados, así como de los objetos de información y de las clases que los implementan, se van recogiendo en un glosario, que permitirá mantener las correspondientes relaciones de trazabilidad entre los diferentes artefactos del modelado.

El Proceso Unificado de desarrollo de software (UP) definido por Rational para UML [6], incluye también el modelado del negocio como un paso más dentro de las iteraciones necesarias que conforman el modelo del proceso. Jacobson et al.[6] presentan algunos pasos que son similares a los nuestros, pero no se considera la descomposición jerárquica de los casos de uso de nivel superior, ni tampoco se proporciona una guía clara para detectar los casos de uso del sistema. Nuestro enfoque para el modelado del negocio constituye una guía completa y detallada, a diferencia de las indicaciones generales presentadas en el UP.

Referencias

1. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley (1999)
2. Ceri, S., Fraternali, P.: Designing Database Applications With Objects and Rules. The IDEA Methodology. Addison-Wesley (1997)
3. Cockburn, A.: Using Goal-Based Use Cases. JOOP, Vol. 10, No. 7 (Nov/Dec 1997) 56-62
4. Coleman, D.: A Use Case Template: Draft for discussion.
http://www.bredemeyer.com/use_case.pdf. (1998)
5. Integration Definition for Function Modeling. Computer Systems Laboratory, National Institute of Standards and Technology, FIPS Pub. 183. December 21 (1993)
6. Jacobson, I., Booch, G. Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley Longman, Inc. (1999)
7. Korson, T.: Misuse of Use Cases.
<http://software-architects.com/publications/korson/korson9803om.htm>. (1998)
8. Korson, T.: Constructing Useful Use Cases.
<http://software-architects.com/publications/korson/usecase3>. (1999)
9. Martin, J. Odell, J.J.: Object-Oriented Methods: A Foundation. Prentice Hall. (1997)
10. Ortín, M.J., García-Molina, J., Martínez, A., Pellicer, A.: Combining OOram and IDEA for Information Systems Modeling. Technical Report TR LSI 1-00. Facultad de Informática. Universidad de Murcia. (December 1998).
11. Ortín, M.J., García-Molina, J.: Modelado con Roles en UML. IV Jornadas de Ingeniería del Software y Bases de Datos. Cáceres, Spain (1999)
12. Reenskaug, T.: Working with Objects: the OOram Software Engineering Method. Addison-Wesley / Manning Publications. (1996)
13. Reenskaug, T.: "Working with Objects: a Three-Model Architecture for the Analysis of Information Systems. JOOP Vol. 10, No. 2 (May 1997) 22-30