

Casos de uso como técnica para especificar el comportamiento de un sistema

*Alberto Restrepo Velásquez**

(Recibido el 4 de diciembre de 2001)

Resumen

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente en [1] y que actualmente forma parte de la propuesta de UML [2].

Los casos de uso están incluidos en el lenguaje unificado de modelamiento (UML) y se utilizan completamente en el Rational Unified Process. Hoy en día han alcanzado gran aceptación en muchos negocios e industrias. Más aún, los casos de uso se aplican a proyectos de software y a una gran variedad de aplicaciones industriales. Se utilizan para especificar los requisitos externamente visibles de un sistema. Se usan en la fase de ingeniería de requisitos y contribuyen a establecer planes de chequeo y guías de usuario. Adicionalmente, se utilizan para crear y validar un diseño propuesto y asegurar que este diseño se ajusta a todos los requisitos.

En este artículo se quiere presentar un panorama acerca de lo que son los casos de uso, sus componentes y cómo identificarlos, cómo se relacionan, cómo se identifican y como aplicarlos en un ejemplo concreto.

----- *Palabras clave:* requisito, actor, escenario, inclusión, extensión, generalización.

Use cases as a technique for system behavior specification

Abstract

Use cases are a functional requirements specification technique. Initially, it was proposed in [1] and recently became part of UML [2].

Use cases are included in the Unified Modeling Language(UML) and are used completely in the Rational Unified Process. Nowadays, use cases have reached great acceptance in many business and industries. Moreover, use cases are applied in software projects and a great variety of industrial applications. They

* Profesor Titular Departamento de Informática y Sistemas. Universidad EAFIT. arestrep@eafit.edu.co.

are used for external visible requirement specifications of a system. They are used in the Requirements Engineering phase and contributed to establish test plans and user guides. Additionally, they are used to create and validate proposed design and asses that the design fixes all requirements.

This article aims at presenting an overview about use cases, their components and how to related them, how identify them and how to apply use case in a concrete example.

----- *Key words:* requirement, actor, scenario, include, extends, generalization.

Introducción

Antes de definir lo que es un caso de uso, es conveniente definir un escenario, tal como aparece en [3]: “Un escenario es una secuencia de pasos que describen una interacción entre un usuario y un sistema”. Por ejemplo, si se tiene un almacén en línea basado en la web, se puede tener un escenario Comprar un producto que diría lo siguiente:

El cliente busca el catálogo y agrega los artículos deseados a la canasta de compras. Cuando el cliente desea pagar, describe la dirección de envío y la información de la tarjeta de crédito y confirma la venta. El sistema chequea la autorización para la tarjeta de crédito y confirma la venta y envía un e-mail de seguimiento.

El escenario es algo que puede suceder. Sin embargo, la verificación de la tarjeta de crédito puede dar negativa. Esto daría para un escenario aparte.

Luego, el caso de uso es un conjunto de escenarios unidos por un objetivo común de un usuario.

El caso de uso ha llegado a ser un término muy popular en el desarrollo orientado a objetos, donde se le define como: “La especificación de una secuencia de acciones, incluyendo variantes, que un sistema u otra entidad puede realizar, interactuando con actores del sistema” [4].

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables.

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo [5].

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requisitos funcionales [6], ya que facilitan el descubrimiento de requisitos y son fácilmente

comprensibles por los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios [7].

A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su utilización concreta [8]. En este documento se propone la utilización de los casos de uso como técnica tanto de descubrimiento/negociación como de especificación de los requisitos funcionales del sistema. Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo las propuestas de [8, 5 y 9] y se describen usando lenguaje natural en forma de patrones.

Diagramas de casos de uso

En estos diagramas, los actores se representan en forma de pequeños muñecos. En la figura 1 se muestra la representación de los actores.



Figura 1 Diagramas de casos de uso [2]

En la figura 2 los casos de uso se representan por elipses.

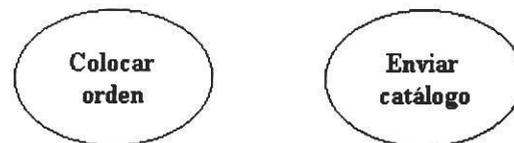


Figura 2 Representación de los casos de uso

La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Un ejemplo de este tipo de diagramas puede verse en la figura 3.

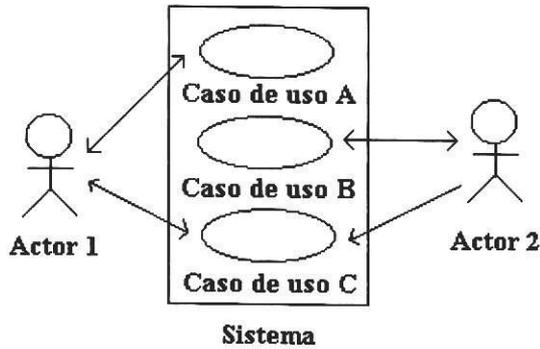


Figura 3 Diagrama de casos de uso

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

Relaciones entre casos de uso

A veces conviene establecer relaciones entre distintos casos de uso para simplificar su descripción. Las dos relaciones posibles y sus semánticas según [2] son las siguientes, cuya representación gráfica puede verse en la figura 4.

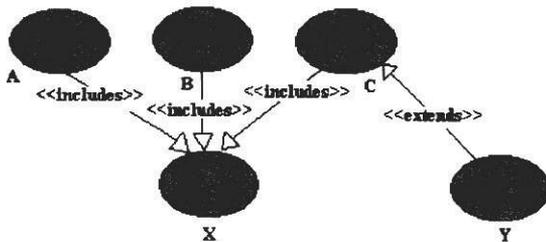


Figura 4 Representación gráfica de relaciones entre casos de uso

Las relaciones *includes* y *extends* se caracterizan por lo siguiente:

- **Includes:** se dice que un caso de uso A *incluye* el caso de uso B, cuando B es una parte del caso de uso A, es decir, la secuencia de interacciones de B forma parte de la secuencia de interacciones de A. También “una relación de un caso de uso a otro que especifica cómo el comportamiento por el primer caso de uso se puede insertar en el comportamiento definido para el segundo caso de uso” [4].

El caso de uso B se realiza siempre dentro del caso de uso A. Además, siempre que ocurre A ocurre también B, por lo que se dice que B es un caso de uso *abstracto* [1, 6].

Un caso de uso es *abstracto* si no puede ser realizado por sí mismo, por lo que sólo tiene significado cuando se utiliza para describir alguna funcionalidad que es común a otros casos de uso. Por otra parte, un caso de uso será *concreto* si puede ser iniciado por un actor y realizado por sí mismo.

Se acostumbra utilizar esta relación cuando se detectan subsecuencias de interacciones comunes a varios casos de uso. Dichas subsecuencias comunes se sacan como “factor común” de los casos de uso que las contienen y se les da forma de casos de uso que son *incluidos* por los casos de uso de los que se han “extraído”. De esta forma se evita repetir las mismas subsecuencias de interacciones una y otra vez en varios casos de uso.

La inclusión según [10], también se conoce como usar un caso de uso. Agrega que el término incluir tiene dos ventajas. La primera es más clara: los pasos de un caso de uso incluyen los de otro. La segunda, se evita la confusión potencial de las palabras “usar” y “uso” en un contexto tan estrecho.

A su vez, [3] utiliza *generalización* cuando se tiene un caso de uso que es similar a otro pero que hace un poco más.

- Extends: un caso de uso A *extiende* a otro caso de uso B cuando A es una subsecuencia de interacciones de B que ocurre en una determinada circunstancia. En [4], “una relación de un caso de uso a otro en la cual el comportamiento definido por el anterior caso de uso emplea el comportamiento definido para el último”.

En cierta forma, A completa la funcionalidad de B. El caso de uso A puede realizarse o no cuando se realiza el caso de uso B, según se den las circunstancias. Por otro lado, el caso de uso A puede ser un caso de uso abstracto o concreto, en cuyo caso puede ocurrir sin necesidad de que ocurra el caso de uso B.

Identificación de las fronteras del sistema

Un paso importante es la definición clara de las fronteras del sistema. Esto significa determinar qué cosas están dentro del sistema y qué cosas están fuera del mismo. Algunas fronteras son fáciles de definir, tal como una persona interactuando con un programa. La forma más sencilla de encontrar las fronteras del sistema, es identificar los actores y los casos de uso.

Identificación de los actores

Los actores son cualquier cosa que interactúa con el sistema. Algunos ejemplos son personas, otros software, dispositivos de hardware, almacenamiento de datos, redes. Cada actor define un papel particular, cada entidad exterior al sistema se puede representar por uno o más actores. Una persona puede representarse por diferentes actores, ya que puede desempeñar diferentes roles. De igual modo, varias personas se pueden representar con un solo actor dado que todos desempeñan el mismo rol.

Los actores son siempre externos al sistema. Ellos no son parte del sistema. Para ayudar a determinarlos, se pueden hacer las siguientes preguntas: [5]

- ¿Quién usa el sistema?
- ¿Quién instala el sistema?
- ¿Cómo inicia el sistema?
- ¿Quién mantiene el sistema?
- ¿Quién apaga el sistema?
- ¿Qué otros sistemas usa el sistema?
- ¿Quién obtiene la información para el sistema?
- ¿Quién provee información al sistema?

Identificación de casos de uso

El siguiente paso es ir a todos los actores e identificar casos de uso para cada uno de ellos. Los casos de uso describen las cosas que los actores quieren que el sistema haga, tal como consultar el estado de una orden existente. En UML, el caso de uso lo inicia un *actor*.

Para lograr la fácil identificación de los casos de uso, se pueden utilizar preguntas como estas:

- ¿Qué funciones querrá el actor del sistema?
- ¿El sistema almacena información?
- ¿Qué actores la crearán, leerán, actualizarán, o borrarán esa información?
- ¿El sistema necesita notificar a un actor acerca de los cambios en su estado interno?
- ¿Hay algún evento externo que el sistema deba conocer?

Cuando considere estas y otras preguntas, determine si esas funciones describen la completa funcionalidad del sistema desde el punto de vista del usuario.

Organización de casos de uso

En la mayoría de los sistemas, el número de casos de uso es lo suficientemente elevado como para que sea oportuno organizarlos de alguna

forma en lugar de tener una *lista plana* por la que no es fácil navegar.

Una posible forma de organizar los casos de uso es recurrir a los *paquetes* descritos en la propuesta de UML [2]. De esta forma, los casos de uso pueden organizarse en niveles, facilitando así su comprensión. Cada paquete contiene a otros paquetes o a varios casos de uso.

Si los casos de uso se agrupan por criterios funcionales, los paquetes que los agrupan pueden tratarse como *subsistemas* [5], tal como puede verse en el ejemplo de la figura 5.

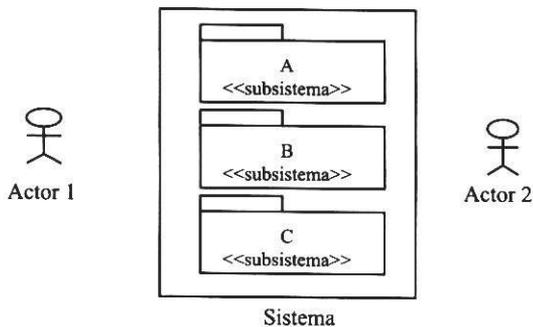


Figura 5 Subsistemas

Un ejemplo

Supóngase que se va a diseñar una máquina despachadora de gaseosas. La función principal es permitir a un cliente adquirir una lata de gaseosa.

¿Cuáles son los posibles casos de uso?

El más obvio es *Comprar gaseosa*. En este caso de uso, es un cliente (actor) que desea comprar una lata de gaseosa. El escenario iniciará cuando el cliente inserte el dinero, haga la selección, y si todo funciona bien, la máquina contará con, por lo menos, una lata de gaseosa que pondrá al alcance del cliente. Se pueden dar otros escenarios como el que no haya gaseosa, o el dinero depositado no sea el suficiente.

Otros casos de uso que se dan en este ejemplo son el del proveedor, que debe *Reabastecer* la

máquina y el de recolectar dinero. Dados estos casos de uso, procederemos a representar en la figura 6 mediante un diagrama, el modelo correspondiente.

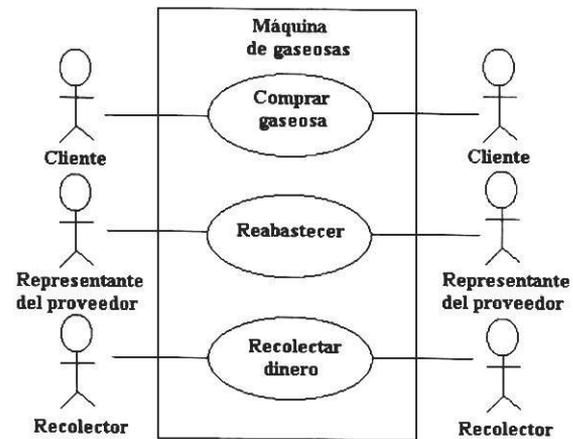


Figura 6 Modelo de máquina de gaseosas

Analícemos ahora, los casos de uso “Reabastecer” y “Recolectar dinero”. Ambos se inician mediante la apertura de la máquina, y finalizan con el cierre y sellado de la misma.

Aquí, podemos tener un caso de uso “Mostrar el interior”, que se crea para capturar el primer par de pasos; y “Cubrir el interior” para el segundo. Como se puede ver, se incluyen este par de casos de uso. Por tanto, se utiliza el *include*. Adicionalmente, el caso de uso “Reabastecer”, puede ser la base de otro caso de uso: “Reabastecer de acuerdo con las ventas”. Esto es, en vez de reabastecer de tal forma que todas las marcas tengan igual cantidad de latas, se puede reabastecer de acuerdo con las que se venden más. Por tanto, se dice que el nuevo caso de uso *extiende* al original dado que agrega otros pasos a la secuencia del caso de uso original, que se conoce como el caso de uso *base*. Esto se puede ver en la figura 7.

Obsérvese que la extensión sólo se puede realizar en los puntos indicados de manera específica dentro de la secuencia del caso de uso base.

Estos puntos se denominan puntos de extensión. Aquí, el punto de extensión aparece debajo del nombre del caso de uso.

Importante, describir los escenarios que se presentan aquí. En [5] se propone la siguiente plantilla para ello:

Scenario document

Scenario Name

<Brief description - usually a paragraph or less>

Use Case

<The use case this scenario is part of>

Priority

<How important is this scenario to the project?>

Status

<What points are we in developing this scenario?>

Flow of Events

<This could be a basic path and alternative paths, or the primary scenario>

Activity Diagram

<An activity diagram of the flow of events, or some significant or complex part of the flow of events>

Sequence Diagrams

<A sequence diagram of the flow of events of the scenario>

Other Artifacts

This can include references to an analysis model, a design model, code, or test plans>

En este ejemplo vamos a describir el flujo de eventos solamente, sin la inclusión de los casos de uso. Posteriormente, aclararemos qué se hace para incluirlos.

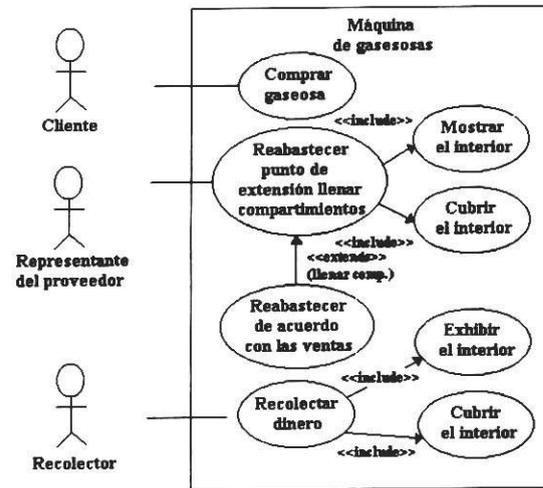


Figura 7 Modelo detallado de máquina de gaseosas

Comprar gaseosa

Flujo de eventos

1. El caso de uso se inicia cuando el cliente inserta el dinero.
2. El cliente hace una selección.
3. La máquina dispensa una lata de gaseosa.
4. El cliente toma la lata de gaseosa y el caso de uso termina

Reabastecer

Flujo de eventos

1. El proveedor inicia el caso de uso.
2. El representante del proveedor quita el seguro a la máquina.
3. El representante hala la puerta para abrir.
4. El representante llena el compartimiento de cada marca hasta su capacidad.
5. El representante rellena la reserva de moneda fraccionaria.
6. El representante cierra la puerta.

7. El representante pone el seguro.
8. El caso de uso termina.

Recolectar dinero

Flujo de eventos

1. El proveedor inicia el caso de uso.
2. El recolector quita el seguro a la máquina.
3. El recolector hala la puerta para abrir.
4. El recolector saca el dinero de la máquina.
5. El recolector cierra la puerta.
6. El recolector pone seguro.
7. El caso de uso termina.

¿Cómo proceder para la inclusión de los casos de uso?

En los caso de uso “Reabastecer” y “Recolectar dinero”, se distinguen ciertos pasos comunes. Ambos empiezan con quitar el seguro, abrir la puerta, y finalizan con cerrar la puerta. Esta duplicación la podemos eliminar tomando cada secuencia de pasos en común y conformando un caso de uso adicional a partir de ellos. Por ejemplo, se puede combinar “quita el seguro” y “hala la puerta” como “Exhibir el interior”; y “cierra la puerta” y “pone el seguro”, como “Cubrir el interior”.

Así, el caso de uso “Reabastecer”, quedaría:

Reabastecer

Flujo de eventos

1. Exhibir el interior.
2. El representante llena el compartimiento de cada marca hasta su capacidad.
3. El representante rellena la reserva de moneda fraccionaria.

4. Cubrir el interior.
5. El caso de uso termina.

De igual modo, se puede proceder con el otro caso de uso.

Resumen

El caso de uso es un técnica poderosa para obtener los requisitos funcionales. Los casos de uso facilitan la comunicación entre analistas y clientes y usuarios.

El caso de uso es una estructura que sirve para describir la forma como lucirá un sistema para los usuarios potenciales. Es una colección de escenarios iniciada por un actor. Debe dar como resultado algo de valor para el actor que lo inició o para otro.

Es posible que se pueda volver a utilizar un caso de uso. Una forma, *includes*, es utilizar los pasos de un caso de uso como parte de la secuencia de pasos de otro caso de uso. Adicionalmente, *extends* permite crear un nuevo caso de uso mediante la adición de pasos de un caso de uso existente.

Referencias bibliográficas

1. I. Jacobson, M. Christerson, P. Jonsson y G. Övergaard. *Object-Oriented Software Engineering: A Use case Driven Approach*. Addison-Wesley, 4ª. edición, 1993.
2. G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
3. M. Fowler y K. Scott. *UML Distilled. Second Edition*. Addison-Wesley, 2000.
4. D. F. D’Souza y A. C. Wills. *Objects, Components, and Frameworks with UML: The Catalysis Approach*. Addison-Wesley, 1999.
5. K. Weidenhaupt, K. Pohl, M. Jarke, y P. Haumer. *Scenarios in System Delopment: Current Practice*. IEEE Software, 15(2):34-45, marzo/abril, 1998.

6. D. G. Firesmith. Use Cases: the Pros and Cons, 1997. Disponible en <http://www.ksscary.com/article7.htm>.
7. Schmuller, Joseph. *Aprendiendo UML en 24 horas*. Prentice-Hall, 1999.
8. A. Cockburn. "Structuring Use Cases with Goals". *Journal of Object-Oriented Programming*, Septiembre, noviembre, y diciembre, 1997. Disponible en <http://members.aol.com/acockburn/papers/usecases.htm>.
9. D. Coleman. A Use Case Template: Draft for Discussion. Fusion Newslwttter, abril, 1998.
10. G. Scheneider y J. P. Winters. *Applying Use Cases: a Practical Guide*. Addison-Wesley, 1999.