

Una Introducción al UML

El Modelo de Casos de Uso

Autor: Geoffrey Sparks, Sparx Systems, Australia

Traducción: Fernando Pincioli (Solus S.A., Argentina) y Aleksandar Orlic (Craftware Consultores Ltda., Chile)

www.sparxsystems.com.ar - www.sparxsystems.cl

Tabla de Contenidos

TABLA DE CONTENIDOS.....	2
EL MODELO DE CASOS DE USO	3
INTRODUCCIÓN AL UML	3
MODELANDO LA FUNCIONALIDAD DEL SISTEMA CON CASOS DE USO	3
LA NOTACIÓN DE CASO DE USO	4
UN EJEMPLO	10
BROWSE BOOK CATALOGUE	11
ENCONTRAR EL LIBRO POR TÍTULO O POR AUTOR	12
SOLICITAR UN LIBRO INEXISTENTE.....	13
LECTURA RECOMENDADA	15

El Modelo de Casos de Uso

Este artículo describe cómo modelar la funcionalidad del sistema utilizando casos de uso. En el UML, los casos de uso son los principales medios para capturar la funcionalidad del sistema desde la perspectiva del usuario y muchas veces puede remplazar al documento "requisitos funcionales".

Introducción al UML

El Lenguaje Unificado de Modelado (UML) es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software. El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos.

El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. El UML también soporta la idea de extensiones personalizadas a través elementos estereotipados.

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo.

Este artículo se enfoca en la representación funcional de los requisitos en UML utilizando Casos de Uso.

En los libros mencionados en la sección de lectura recomendada se puede encontrar más información sobre el UML y de los documentos de especificación del UML que se pueden encontrar en las paginas de recursos de UML del OMG (*Object Management Group*) www.omg.org/technology/uml/ y www.omg.org/technology/documents/formal.

Modelando la Funcionalidad del Sistema con Casos de Uso

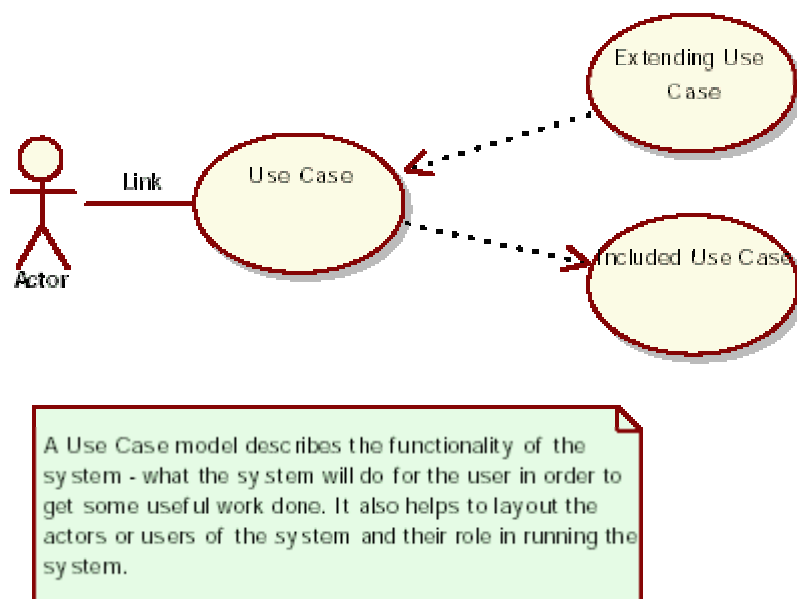
El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un Caso de Uso es una unidad de trabajo significativo; por ejemplo crear una solicitud y modificar una solicitud son todos Casos de Uso.

Cada Caso de Uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto. Un Caso de Uso puede 'incluir' la funcionalidad de otro Caso de Uso o puede 'extender' otro Caso de Uso con su propio comportamiento.

Los casos de uso típicamente se relacionan con 'actores'. Un actor es un humano o una máquina que interactúa con el sistema para realizar un trabajo significativo.

La Notación de Caso de Uso

Los diagramas de están generalmente compuestos por uno o más actores vinculados con uno o más casos de uso, como en el diagrama siguiente.



El Caso de Uso

Un Caso de Uso es una representación de una unidad discreta de trabajo realizada por un usuario (u otro sistema) usando el sistema en operación. Se ejecuta en su totalidad o no se ejecuta nada, devolviendo algo de valor al usuario. Algunos ejemplos de casos de uso son AgregarPedido, EliminarPedido, ModificarPedido, etc.

Una descripción de Caso de Uso generalmente incluirá:

Comentarios generales y notas que describen el Caso de Uso;

Requisitos: cosas que el Caso de Uso debe permitir hacer al usuario, tales como <capacidad de actualizar orden>, <capacidad de modificar orden>, etc.

Restricciones: las reglas sobre qué se puede hacer y qué no se puede. Incluyen precondiciones que tienen que ser verdaderas antes de que se ejecute el Caso de Uso (por ejemplo <crear orden> debe preceder a <modificar orden>); también incluyen Post-condiciones que tienen que ser verdaderas una vez que el Caso de Uso se ejecutó (por

ejemplo <la orden está modificada y es consistente>); Invariantes: son siempre verdaderos (por ejemplo, una orden debe tener siempre un número de cliente).

Escenarios: descripciones secuenciales de los pasos que se llevan a cabo para ejecutar un Caso de Uso. Puede incluir múltiples escenarios para abarcar las circunstancias excepcionales y los caminos de procesamiento alternativos.

Diagramas de escenarios: diagramas de secuencia para representar el flujo; semejante al 4, pero plasmado gráficamente.

Actores

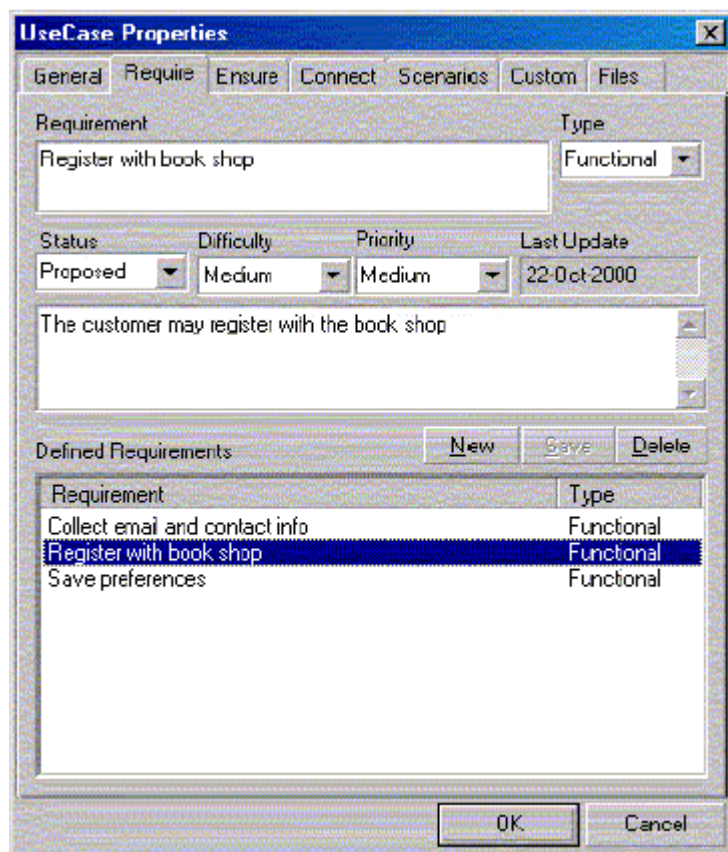


Un actor es un usuario del sistema. Esto incluye usuarios humanos y otros sistemas computacionales. Un actor usa un Caso de Uso para ejecutar una porción de trabajo de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define rol en el sistema y el alcance de su acción.

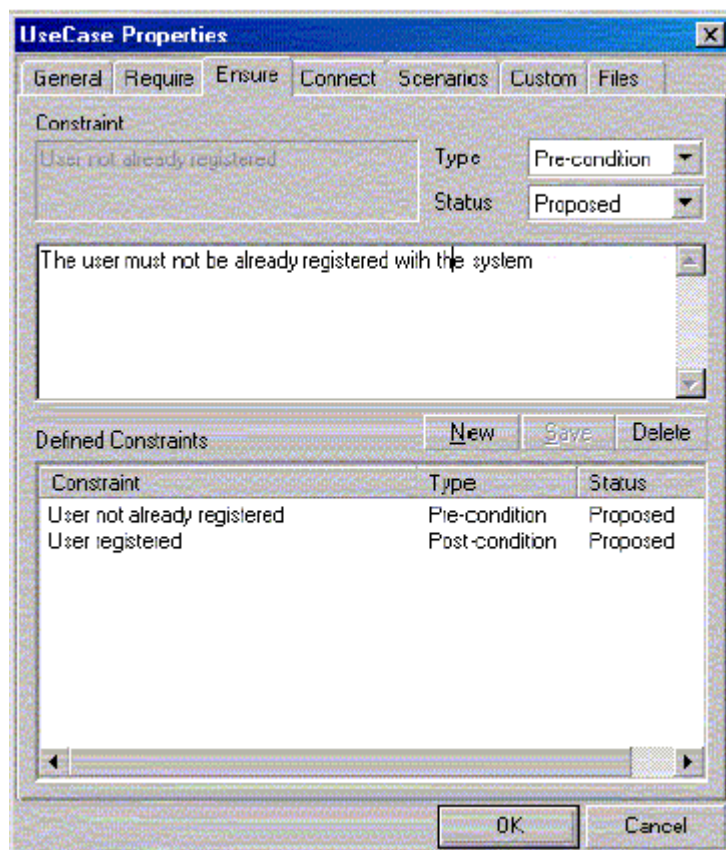
Restricciones, requisitos y escenarios

La especificación formal de un Caso de Uso incluye 3 elementos básicos:

Requisitos. Son los requisitos funcionales formales que el Caso de Uso debe proveer al usuario final. Ellos corresponden a las especificaciones funcionales de las metodologías estructuradas. Un requisito es un contrato de que el Caso de Uso realizará alguna acción o proveerá algún valor al sistema.



Restricciones. Estas son las reglas formales y las limitaciones bajo las que opera un Caso de Uso e incluyen las pre-condiciones, las post-condiciones y las invariantes. Una precondición especifica qué debe haber ocurrido o estar cumplido antes de que el Caso de Uso pueda iniciarse. Una post-condición documenta qué será verdadero una vez que el Caso de Uso se complete. Una invariante especifica qué será verdadero durante el tiempo en que opere el Caso de Uso.



Escenarios. Son descripciones formales del flujo de eventos que ocurre durante una instancia de un Caso de Uso. Usualmente se describen con texto y corresponden a una representación textual del diagrama de secuencia.

Incluye y Extiende

Un Caso de Uso puede incluir la funcionalidad de otro como parte de su procesamiento normal. Generalmente se asume que los casos de uso incluidos se llamarán cada vez que se ejecute el camino base. Un ejemplo puede ser listar un conjunto de órdenes de clientes de las cuáles poder elegir antes de modificar una orden seleccionada; en este caso, el Caso de Uso <listar órdenes> se puede incluir en el Caso de Uso <modificar orden> cada vez que éste se ejecute.

Un Caso de Uso puede ser incluido por uno o más casos de uso, ayudando así a reducir la duplicación de funcionalidad al factorizar el comportamiento común en los casos de uso que se reutilizan muchas veces.

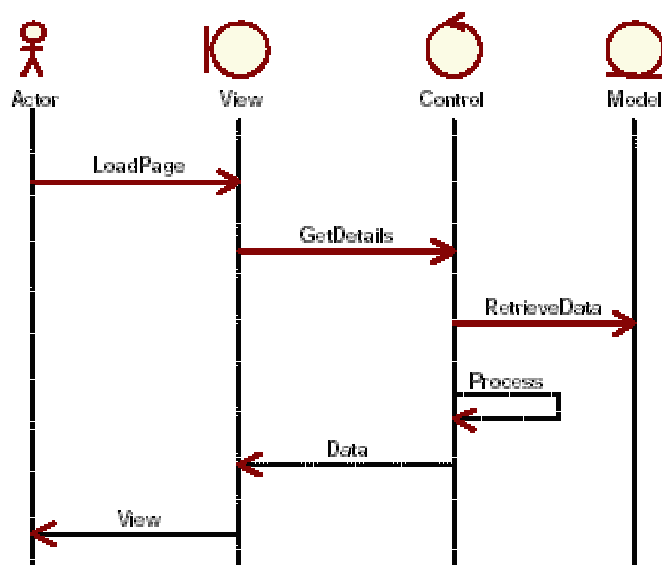
Un Caso de Uso puede extender el comportamiento de otro Caso de Uso; típicamente cuando ocurren situaciones excepcionales. Por ejemplo, si antes de modificar un tipo particular de orden de cliente, un usuario debe obtener la aprobación de alguna autoridad superior, entonces el Caso de Uso <obtener aprobación> puede extender opcionalmente el Caso de Uso normal <modificar orden>.

Diagramas de Secuencia

El UML provee un medio gráfico para representar la interacción entre los objetos a lo largo del tiempo en los diagramas de secuencia. Éstos muestran típicamente a un usuario o a un actor y los objetos y componentes con los que interactúen durante la ejecución de un Caso de Uso. Un diagrama de secuencia representa típicamente un único escenario de Caso de Uso o flujo de eventos.

Los diagramas son una vía excelente para documentar los escenarios de uso, para capturar los objetos necesarios de manera temprana en el análisis y para verificar el uso de los objetos más tarde en el diseño. Los diagramas de secuencia muestran el flujo de mensajes de un objeto a otro y, como tales, representan los métodos y los eventos soportados por un/a objeto/clase.

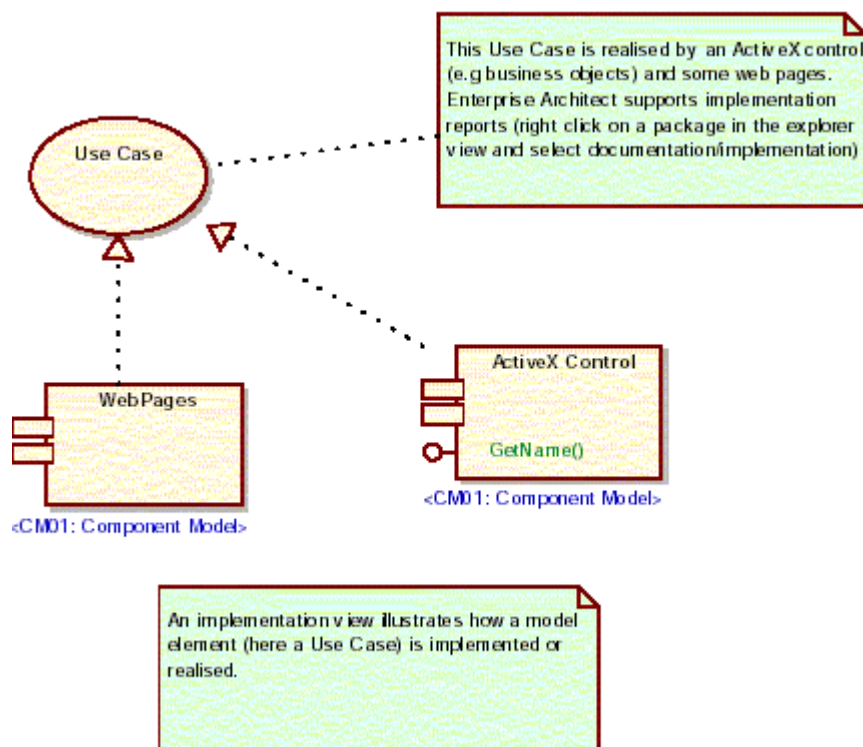
El diagrama siguiente muestra un ejemplo de diagrama de secuencia con el usuario o actor a la izquierda iniciando un flujo de eventos y mensajes que corresponden al escenario del Caso de Uso.



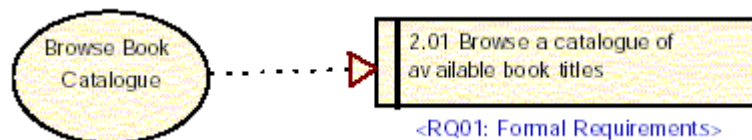
A workflow or sequence diagram may be anchored under a Use Case to further illustrate the sequence of steps taken to realise a use case. The degree of detail here depends on what it is desired to communicate - from the simple interaction of a user and the user interface to the more complicated interaction of objects and controllers within the system and with other systems.

Diagrama de Implementación

Un Caso de Uso es una descripción formal de la funcionalidad que el sistema tendrá cuando se construya. Un diagrama de implementación se asocia típicamente con un caso de uso para documentar qué elementos de diseño (por ejemplo, componentes y clases) implementará la funcionalidad del Caso de Uso en el nuevo sistema. Esto provee un alto grado de trazabilidad al diseñador, al cliente y al equipo que construirá el sistema. La lista de casos de uso a los que se asocia un componente o una clase documenta la funcionalidad mínima que debe ser implementada por el componente.

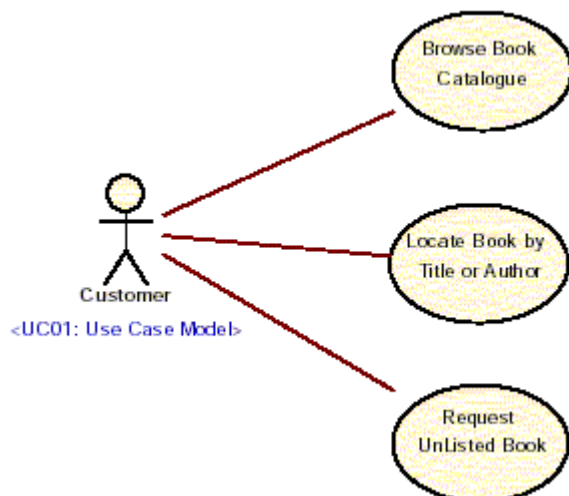


Se puede construir un segundo tipo de Diagrama de Implementación para ilustrar qué requisitos formales implementa o realiza un Caso de Uso. El ejemplo siguiente muestra que el Caso de Uso "Recorrer el Catálogo de Libros" ("Browse Book Catalogue") implementa el requisito formal "2.01 Recorrer un catálogo de títulos de libros disponibles" ("2.01 Browse a catalogue of available book titles"). Este segundo tipo de diagrama no es estrictamente un diagrama UML pero puede ser una herramienta muy importante para documentar cómo cumple el sistema propuesto las especificaciones establecidas por un cliente.



Un Ejemplo

Este diagrama ilustra los casos de uso que soportan la búsqueda de un libro y el recorrido de un conjunto de registros resultante. El cliente puede ingresar un criterio de búsqueda y recorrer los resultados. Si lo desea, el usuario puede seleccionar un ítem para agregarlo a su actual carro de compras.



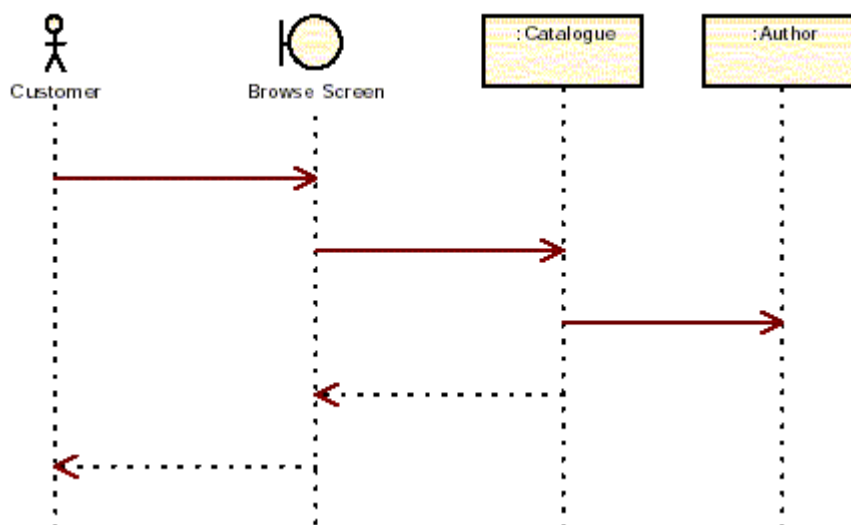
Browse Book Catalogue

El cliente puede navegar en línea el catálogo de libros actual. Éste debería mostrar los detalles y la cantidad disponible de los libros. El usuario debería poder filtrar por título del libro, por autor y por categoría del libro.

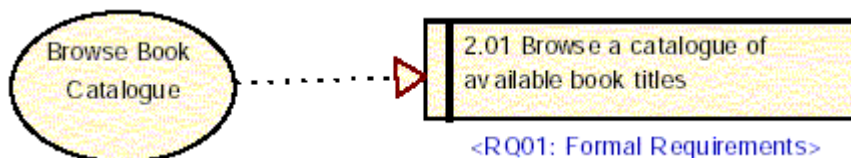
Requisitos

1. *Propuesto*. Mostrar un conjunto de libros filtrado.
2. *Propuesto*. Navegar la lista.
3. *Propuesto*. Pagar el siguiente conjunto de resultados.
4. *Propuesto*. Agregar un ítem al carro de compras.
5. *Propuesto*. Obtener más información acerca de un ítem.

Diagrama de secuencia "Recorrer Catálogo" (*Browse Catalogue*)



Recorrer Catálogo de Libros (Browse Book Catalogue): Implementación



Encontrar el Libro por Título o por Autor

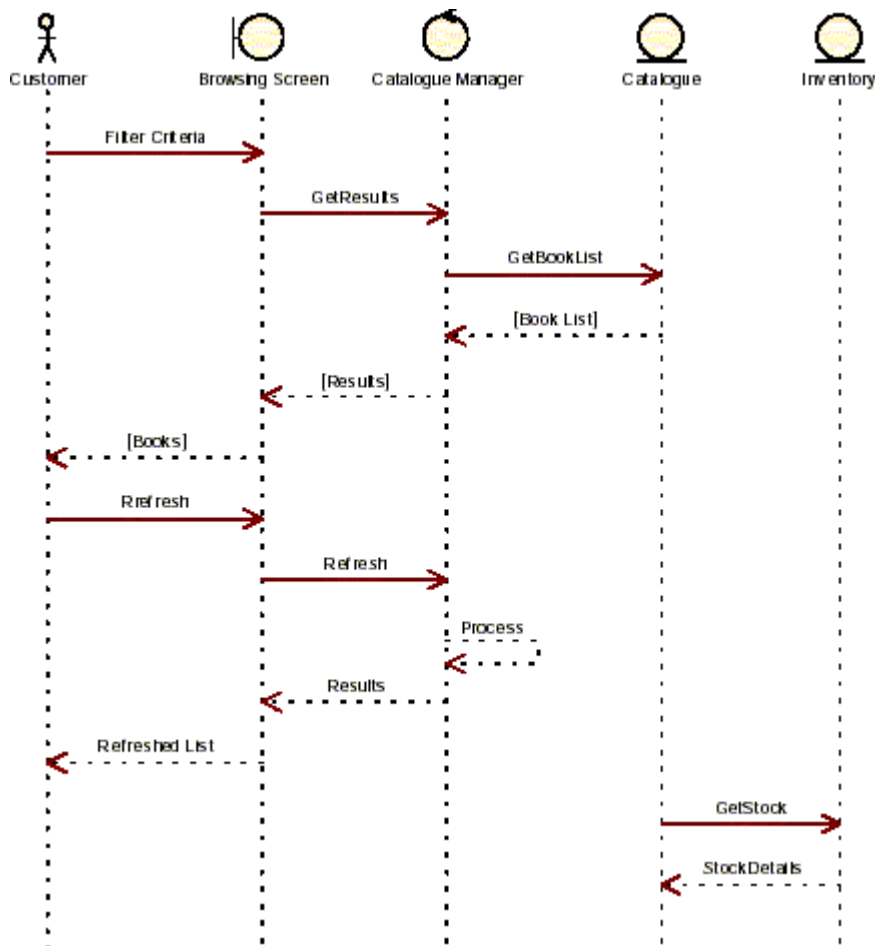
El usuario ingresa algún criterio de filtro, tal como el título del libro o el autor. Esto limita el conjunto de libros obtenidos a aquellos que satisfagan el criterio indicado. Se deberían soportar caracteres comodines tales como "*" y "?".

Requisitos

1. *Propuesto*. Ingresar el criterio de búsqueda por autor.
2. *Propuesto*. Ingresar el criterio de búsqueda de título del libro.
3. *Propuesto*. Actualizar la consulta.

Diagrama de secuencia "Encontrar Libro" (*Locate Book*)

Este diagrama ilustra el flujo de eventos que suceden cuando un usuario busca un libro en línea. El sistema devolverá detalles del catálogo y del inventario para el ítem seleccionado.



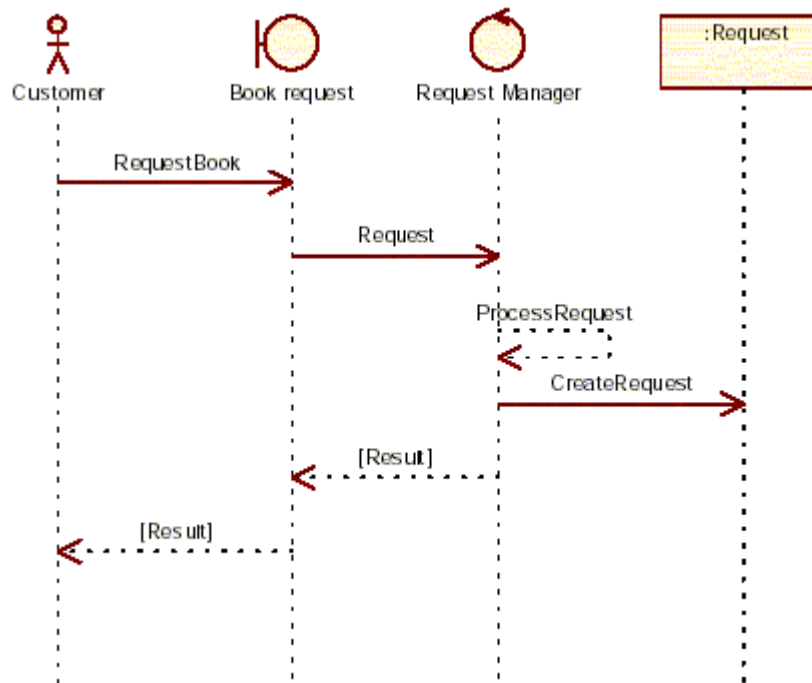
Solicitar un Libro Inexistente

Si el usuario no puede encontrar un libro en la categoría actual, debería poder realizar una solicitud especial. Debería incluir detalles tales como Autor, Editor, Título, ISBN y cualquier otro tipo de información útil.

Requisitos

1. *Propuesto*. Ingresar una solicitud de un libro no existente en el catálogo.

Solicitar un Libro Inexistente (*Request UnListed Book*): Solicitar un Libro (*Request Book*)



Lectura Recomendada

Sinan Si Alhir, **UML in a NutShell**.

ISBN: 1-56592-448-7. Publisher: O'Reilly & Associates, Inc

Doug Rosenberg with Kendall Scott ,Use Case Driven Object Modeling with UML.

ISBN: 0-201-43289-7. Publisher: Addison-Wesley

Geri Scheider, Jason P. Winters, **Applying Use Cases**

ISBN: 0-201-30981-5. Publisher: Addison-Wesley

Ivar Jacobson, Martin Griss, Patrik Jonsson, **Software Reuse**

ISBN: 0-201-92476-5. Publisher: Addison-Wesley

Hans-Erik Eriksson, Magnus Penker, **Business Modeling with UML**

ISBN: 0-471-29551-5. Publisher: John Wiley & Son, Inc

Peter Herzum, Oliver Sims, Business Component Factory

ISBN: 0-471-32760-3 Publisher: John Wiley & Son, Inc