

Lenguaje unificado de modelado: casos de uso y clases y secuencia

Semana 5

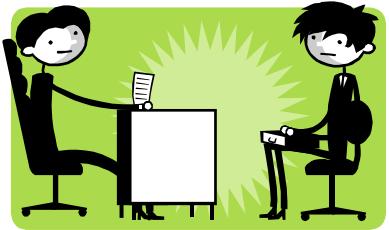
**Agradecimiento: la mayoría del material incluido en
estas diapositivas es una cortesía del Dr. Rami Bahsoon**

Agenda

- Lenguaje unificado de modelado
- Diagramas de casos de uso
- Diagramas de clases
- Diagramas de secuencia

Lenguaje unificado de modelado

A breves rasgos



Extracción de Requerimientos



Especificación de Requerimientos



Proceder



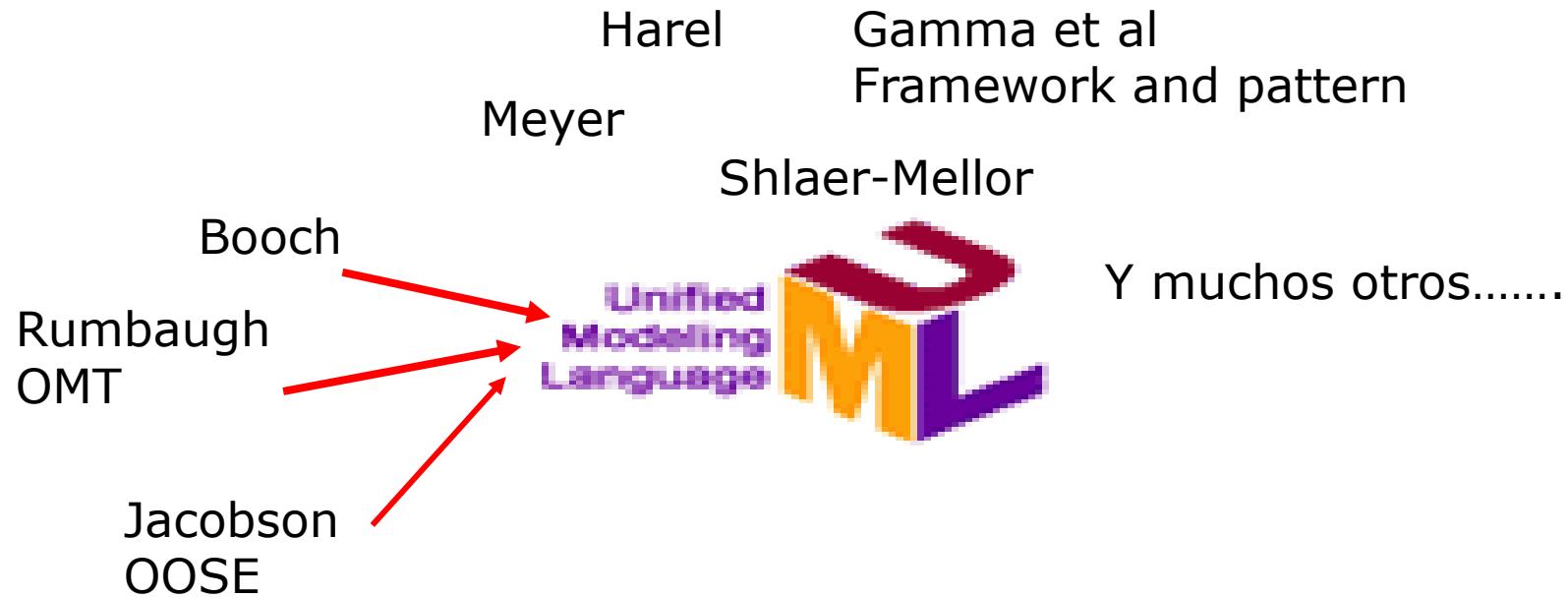
Se pueden hacer usando
UML ;-)

Lenguaje unificado de modelado

- Se propusieron varias notaciones diferentes para describir diseños orientados a objetos en los años ochenta y noventa.
- El lenguaje de modelado unificado es una integración de estas notaciones.
- Describe notaciones para varios modelos diferentes que pueden producirse durante el análisis y diseño de OO.
- Ahora es un estándar **de facto** para el modelado OO.

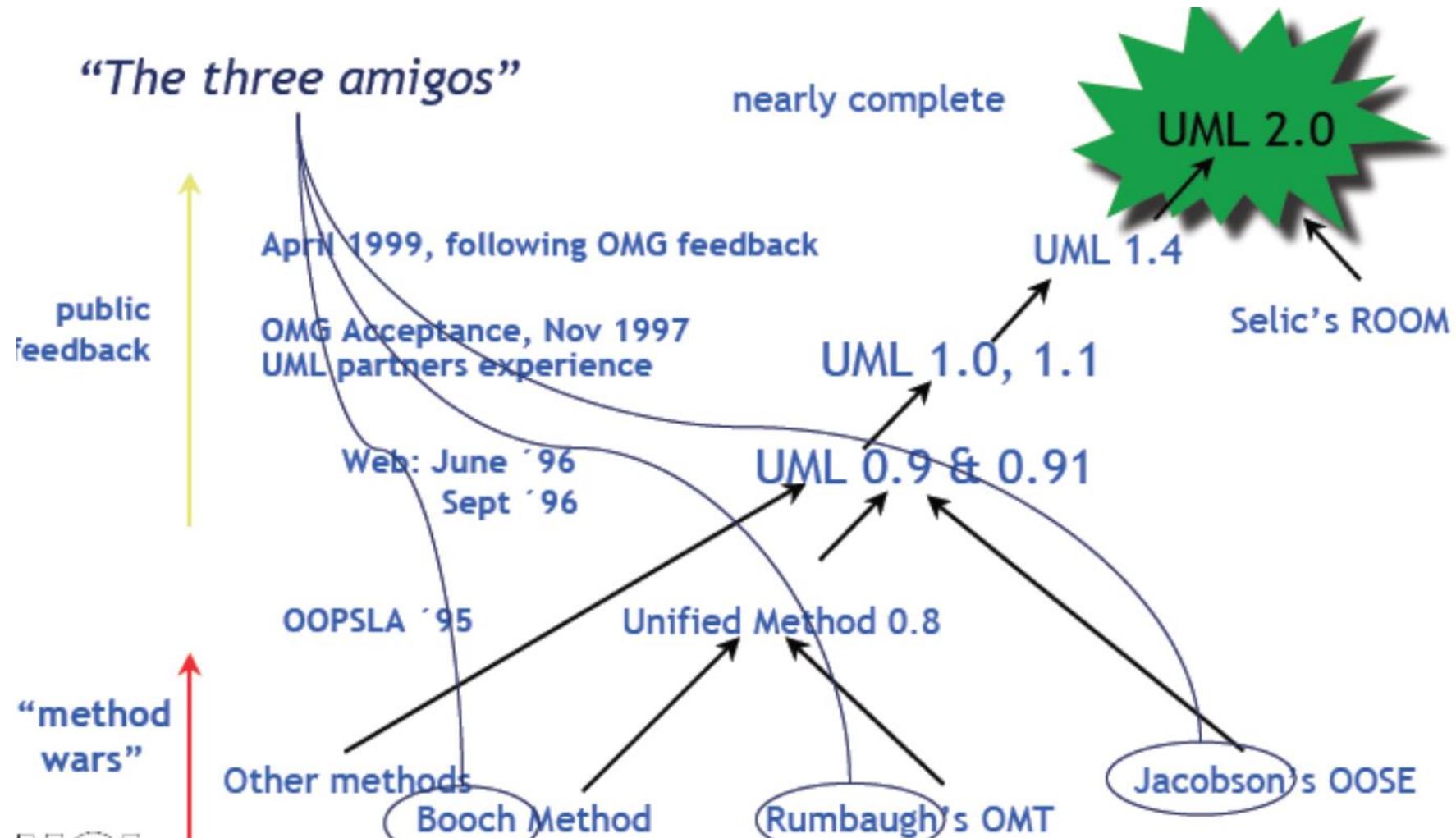
Contribuyentes a UML

- <http://www.uml.org/>

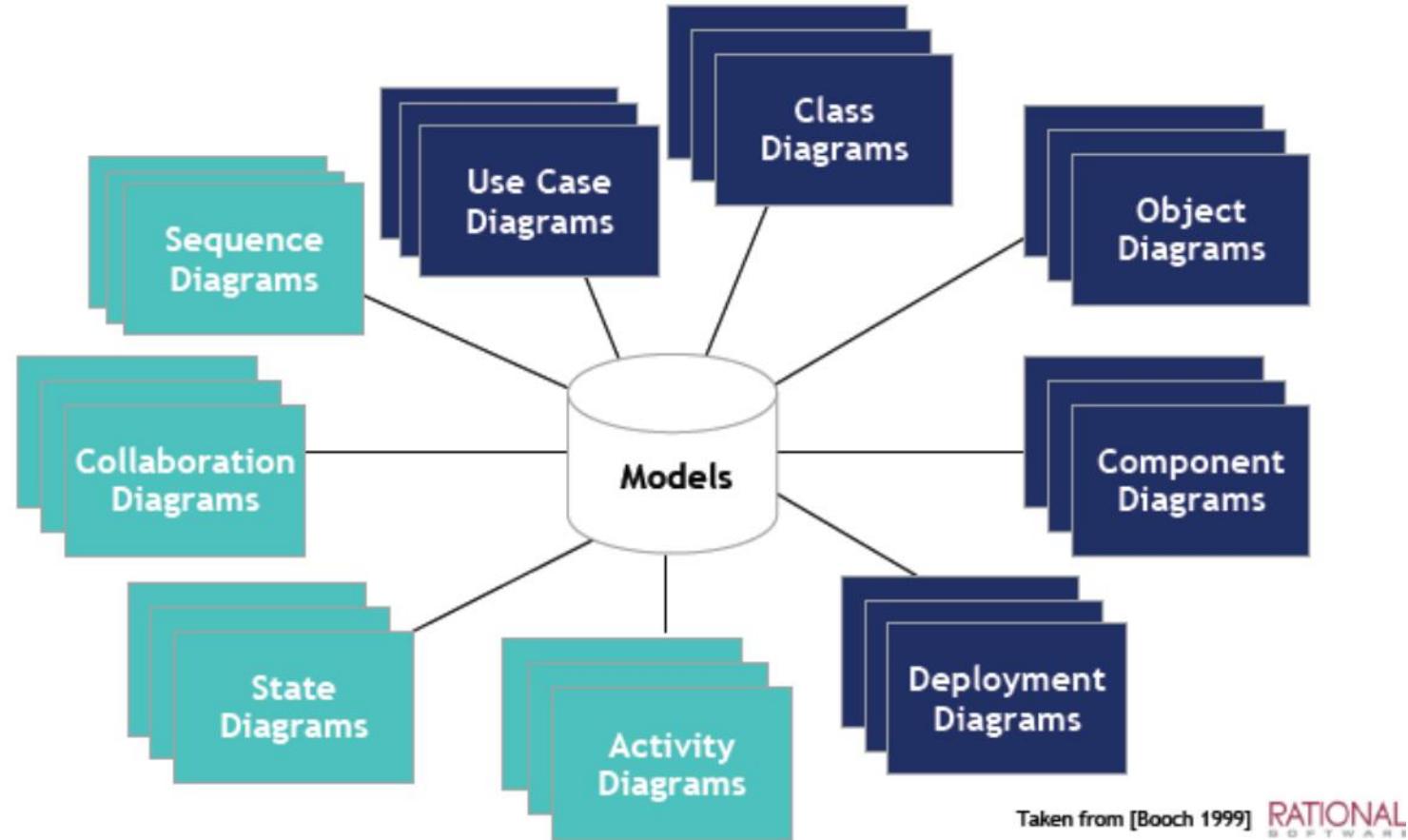


Los tres principales (submission to OMG Jan 97, Acceptance Nov 97...)
<http://www.omg.org/>

Recorrido hasta UML 2.5.1



Diagramas UML



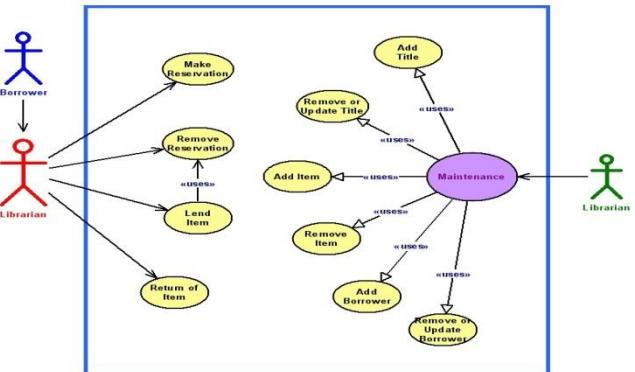
¿Qué es un modelo?

- El lenguaje del diseñador.
- **Representaciones** del sistema ha ser construido o cómo está construido
- Una descripción completa de un sistema desde una perspectiva particular.
- Vehículos para la comunicación con diversos interesados.
- Permitir razonar sobre algunas características de un sistema
- A menudo captura información **estructural** y de **comportamiento** (por ejemplo, interacción)

Diagramas UML

- Diagrama: una vista del modelo
- En UML, hay nueve diagramas estándar.
- **Vista estática**: caso de uso, clase, objeto, componente, implementación
- **Vista dinámica**: secuencia, colaboración, gráfico de estado, actividad

Algunos diagramas UML



Casos de Uso

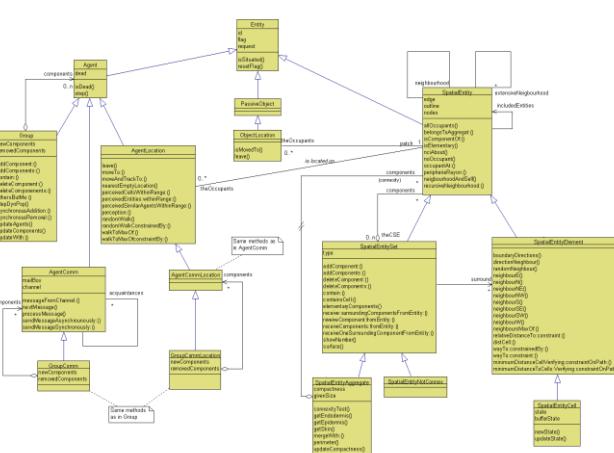
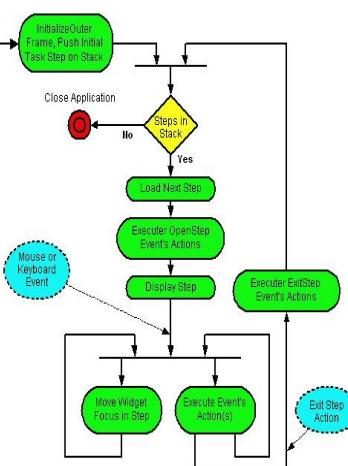
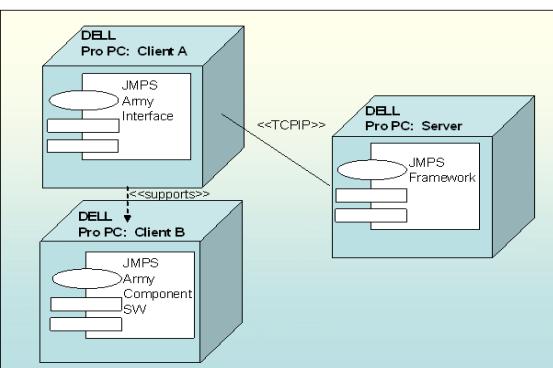


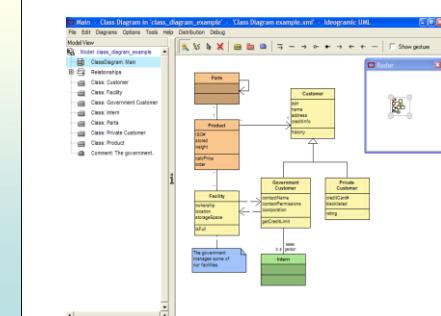
Diagrama de Clases



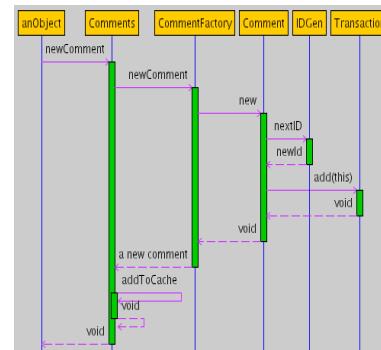
Actividad



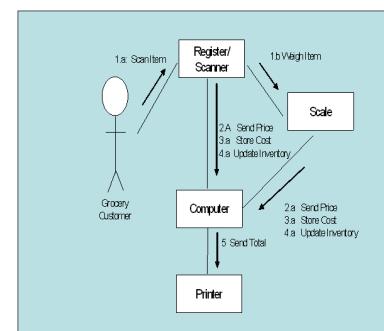
Despliegue



Unified
Modeling
Language



Secuencia

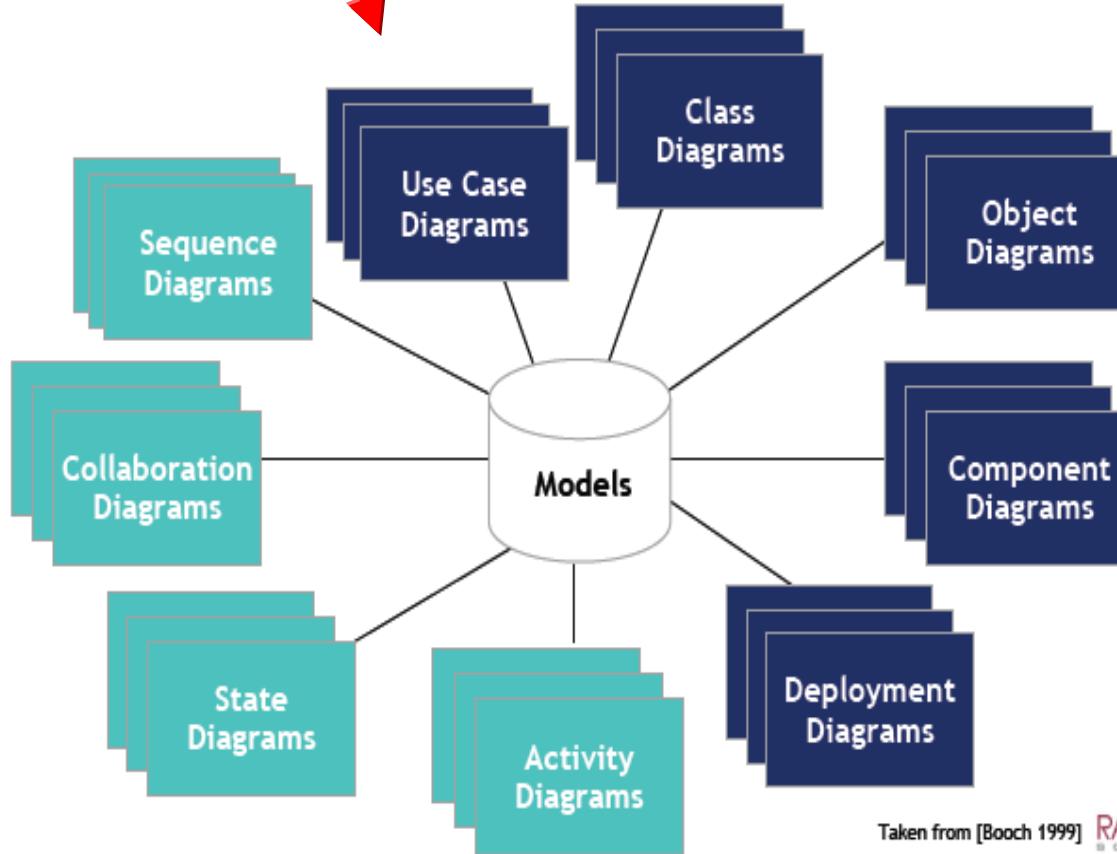


Colaboración

Diagramas de casos de uso

Diagramas UML

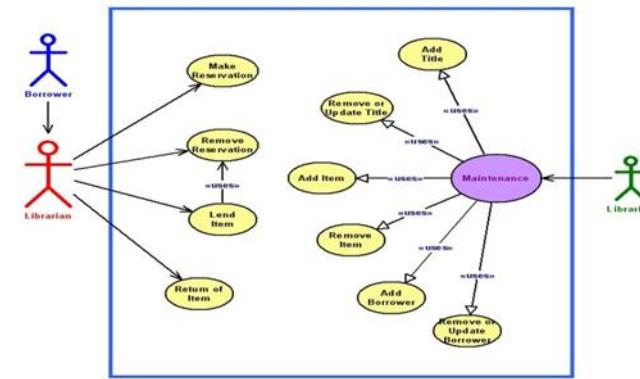
Estamos aquí!



Taken from [Booch 1999] RATIONAL SOFTWARE

Casos de Uso

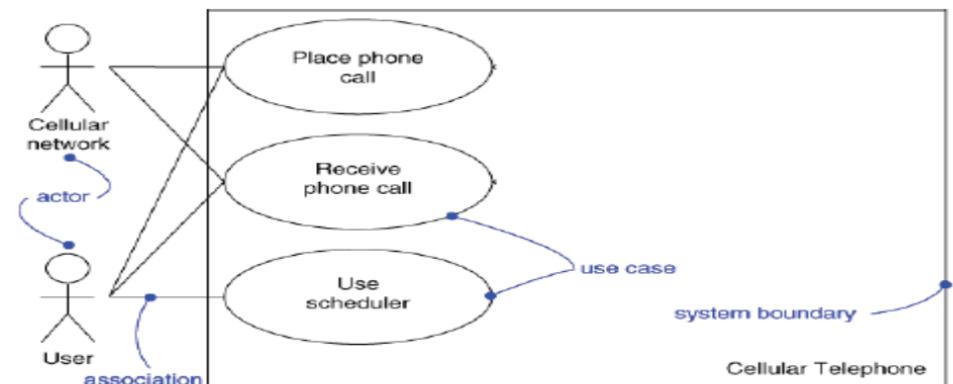
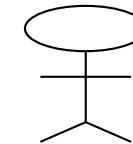
- ¿Qué es el modelado de casos de uso?
- ¿Qué son los actores?
- ¿Cómo encontrar actores?
- ¿Qué son los casos de uso?
- ¿Cómo encontrar casos de uso?
- ¿Cómo construir un diagrama de caso de uso?
- Detallando un caso de uso ...



¿Qué es un modelo de caso de uso?

- Bases de un enfoque orientado al usuario para el desarrollo del sistema

- Identifica los usuarios del sistema (actores)
- Identifica las tareas que deben hacer con el sistema (casos de uso)
- Relaciona usuarios y tareas (relación) ... ayuda a identificar límites



Taken from [Booch 1999] RATIONAL SOFTWARE

- Captura la funcionalidad del sistema tal como la ven los usuarios

Casos de uso

- Construido en las primeras etapas de desarrollo.
 - Especifica el contexto de un sistema.
 - Planifica iteraciones de desarrollo
 - Valida la arquitectura de un sistema
 - Impulsa la implementación y genere casos de prueba
 - Desarrollado por analistas y expertos en dominios durante el análisis de requerimientos.

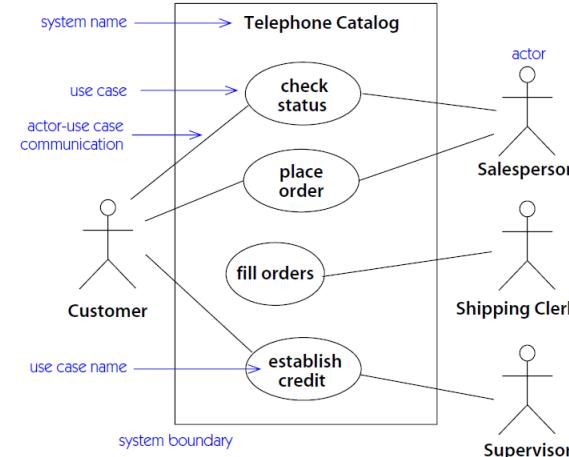
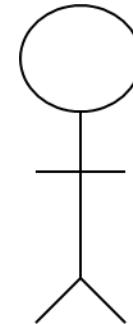


Figure 5-1. Use case diagram

¿Cómo encontrar actores?

- Observar a los usuarios directos del sistema: podrían ser usuarios o sistemas
 - ¿Qué papel juegan ellos?
 - ¿Quién proporciona información al sistema?
 - ¿Quién recibe información del sistema?
- Los actores podrían ser:
 - Principal
 - Secundario (hardware externo, otros sistemas, ...)
- Describa cada actor de manera clara y precisa (semántica)
 - Nombre corto
 - Descripción



BookBorrower

Este actor representa alguien que usa la biblioteca para prestar libros

Ejercicio 1

- Suponga que tiene documentos de requerimientos para un sistema de biblioteca: identifique todos los actores que interactúan con un sistema.
- Para cada actor, escriba el nombre y proporcione una breve descripción textual (es decir, describiendo la semántica del actor)

Actor	Semantics
Name 1	Description

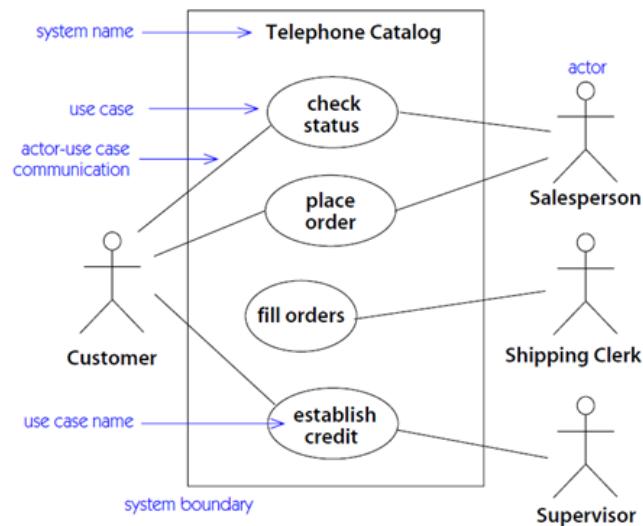


Figure 5-1. Use case diagram

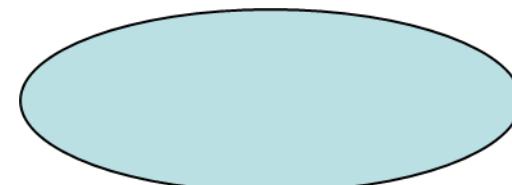
¿Qué son casos de uso?

- Cosas que los actores hacen con el sistema
 - Una tarea que un actor debe realizar con la ayuda de un sistema (por ejemplo, pedir prestado un libro)
 - Una funcionalidad específica de sistema
- Describe el comportamiento del sistema desde el punto de vista del usuario.
- Representado por elipses

Borrow a copy
of book

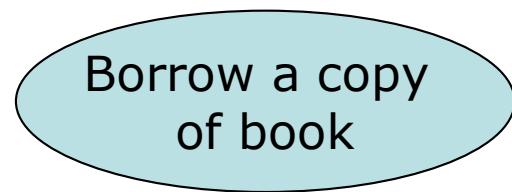
¿Cómo encontrar casos de uso?

- Comience con la lista de actores y considere
 - Lo que necesitan del sistema (es decir, qué casos de uso hay que tengan valor para ellos)
 - Cualquier otra interacción que esperan interactuar con el sistema (es decir, en qué casos de uso podrían participar para el beneficio de otra persona)
- ¿Cómo sabes qué es un caso de uso?
 - Estime la frecuencia de uso, examine las diferencias entre los casos de uso, distinga entre el curso de eventos "básico" y "alternativo" y cree nuevos usos cuando sea necesario



Describiendo casos de uso

- Semántica detallada en el texto.



Debería ser descrito →



- Ejemplo:
- Pedir prestada la copia de un libro
 - Un prestatario de libros presenta un libro. El sistema verifica que el potencial prestatario sea miembro de la biblioteca y que no tenga prestada la cantidad máxima de libros.

Ejercicio

- Elabore el diagrama de casos de uso de un sistema de biblioteca
 - Use Visual Paradigm (ya sea la versión en línea o la community):
<https://www.visual-paradigm.com/download/>

Posibles casos de uso...

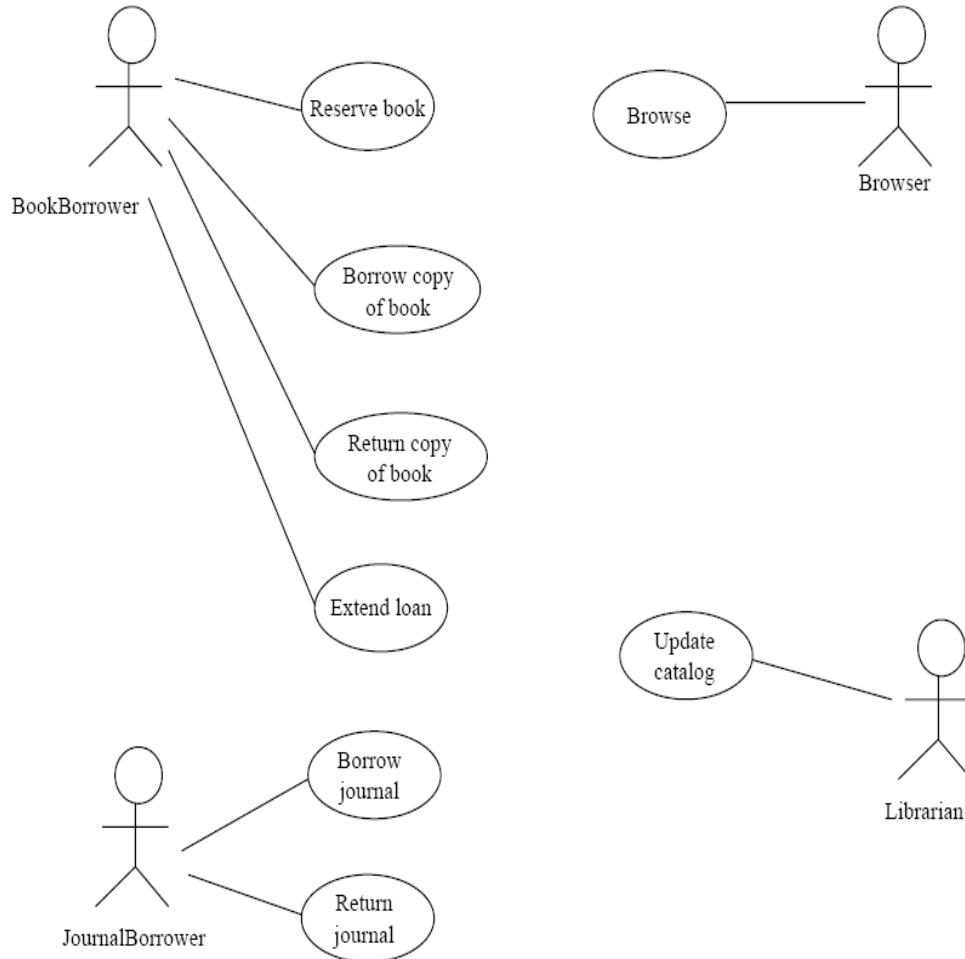
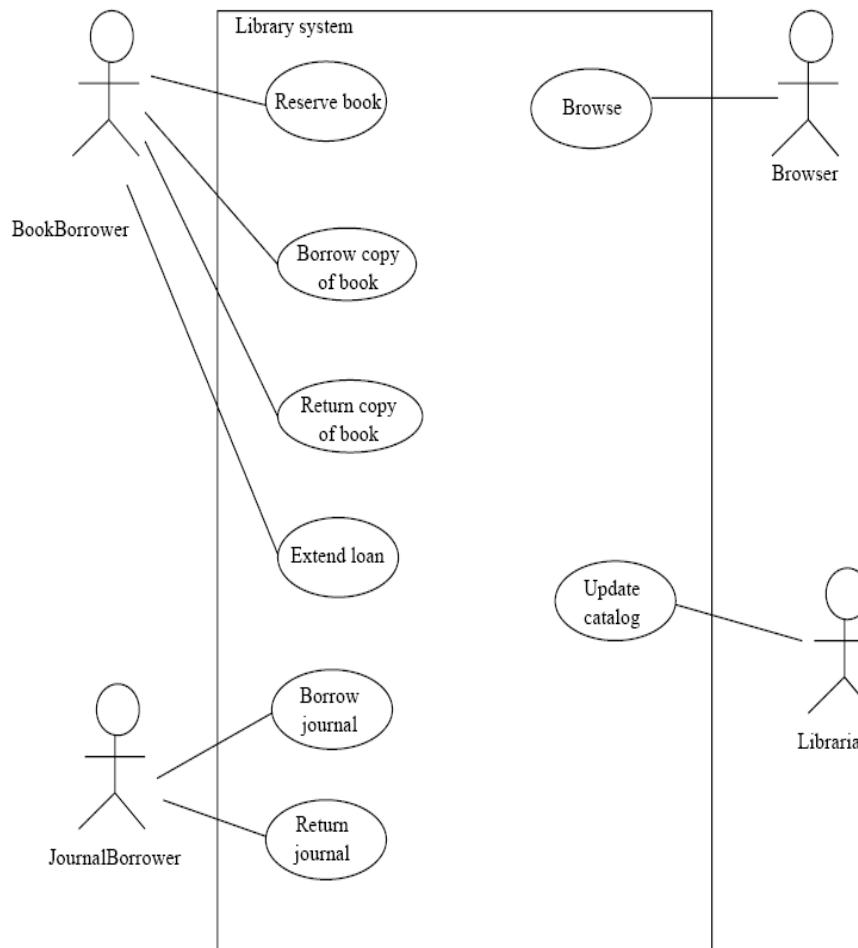


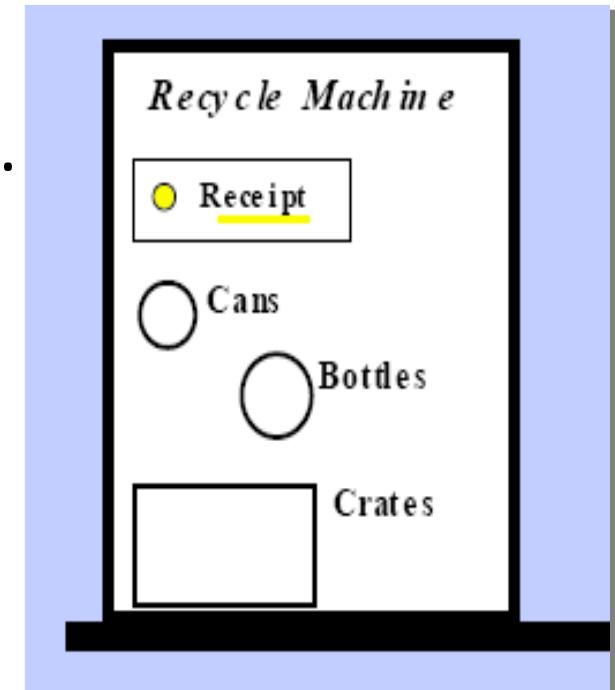
Diagrama de caso de uso de una biblioteca



Ejemplo de requerimientos

- Una máquina de reciclaje multipropósito debe:
 - recibir y verificar artículos para clientes,
 - imprimir recibo de los artículos recibidos,
 - imprimir el total de artículos recibidos para el operador,
 - cambiar la información del sistema,
 - emitir señal de alarma cuando surjan problemas.

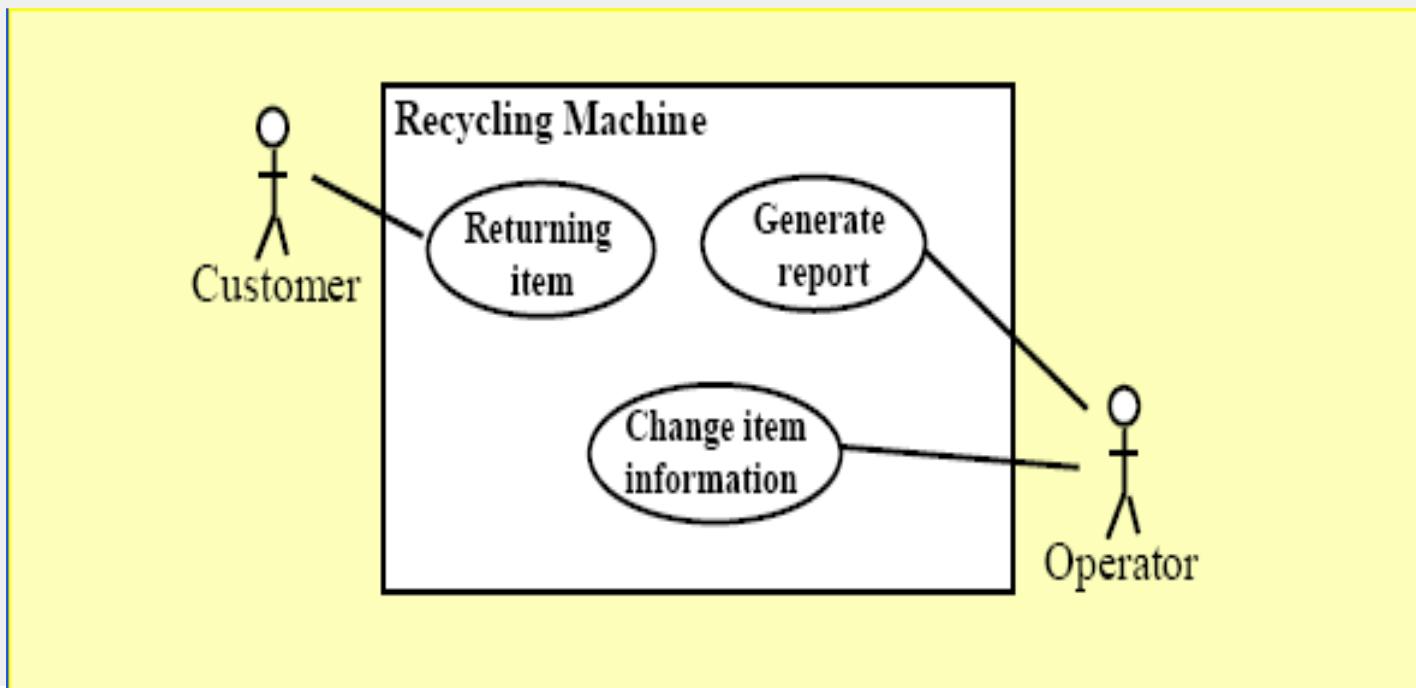
Referencia: Anthony Finkelstein, UCL



Ejemplo

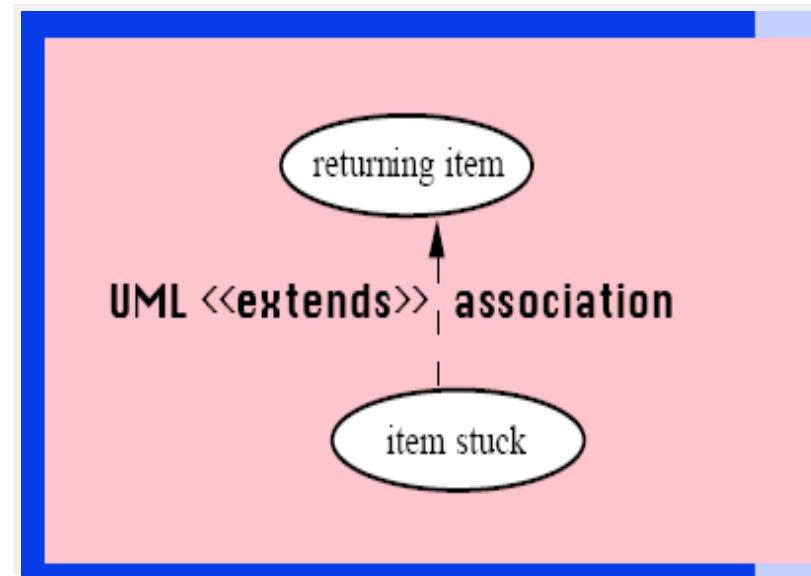
- El retorno de ítems es iniciado por el Cliente cuando este desea retornar latas, botellas o cajas. Con cada ítem que el Cliente coloca en la máquina de reciclaje, el sistema aumentará el número de ítems recibidos del Cliente, así como el total diario de este tipo en particular.
- Cuando el Cliente haya depositado todos sus artículos, presionará un botón de recibo para obtener un recibo en el que los ítems devueltos hayan sido impresos, así como la suma total de devolución.
- Las instancias particulares de uso serían diferentes ...mañana después de la fiesta, Sarah va al centro de reciclaje con tres cajas que contienen ...

Diagrama de caso de uso



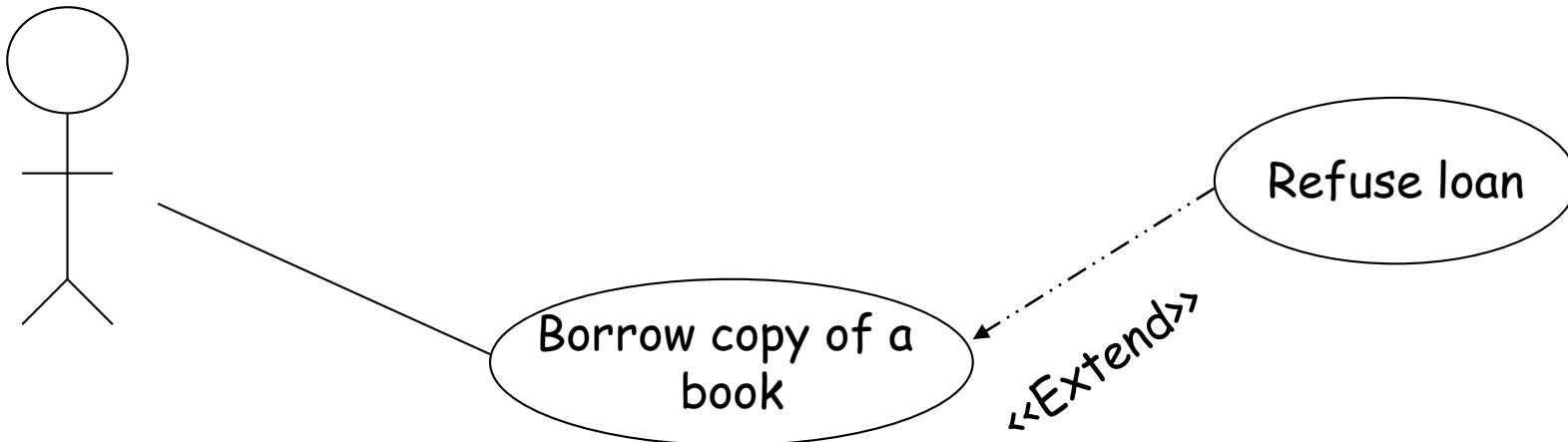
Extensiones

- Las extensiones brindan oportunidades para:
 - partes opcionales
 - casos complejos alternativos
 - sub-casos separados
 - inserción de casos de uso



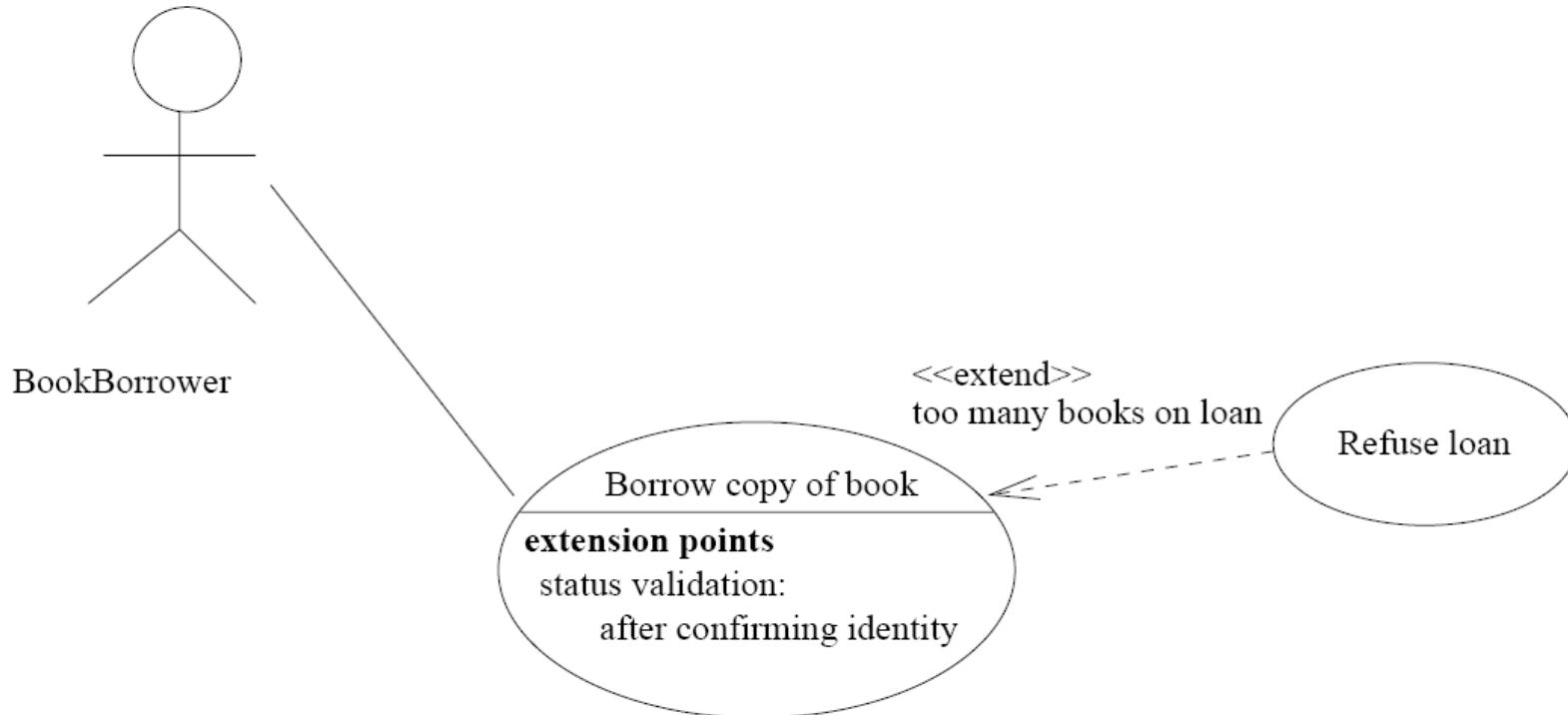
Refinamiento - <<extend>>

- <<extend>>

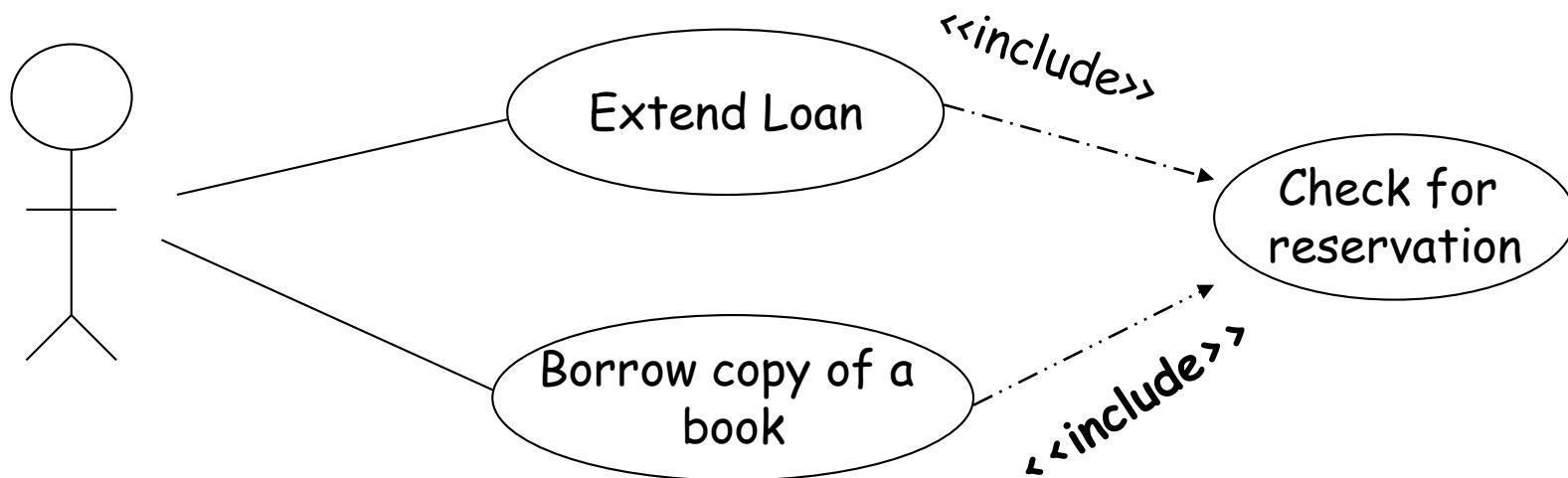


- Nota: la dirección de la flecha desde el caso menos central al central
- **Rechazar préstamo y pedir prestada la copia de un libro en dos escenarios diferentes**

Refinamiento - <<extend>>



Refinamiento - <<include>>

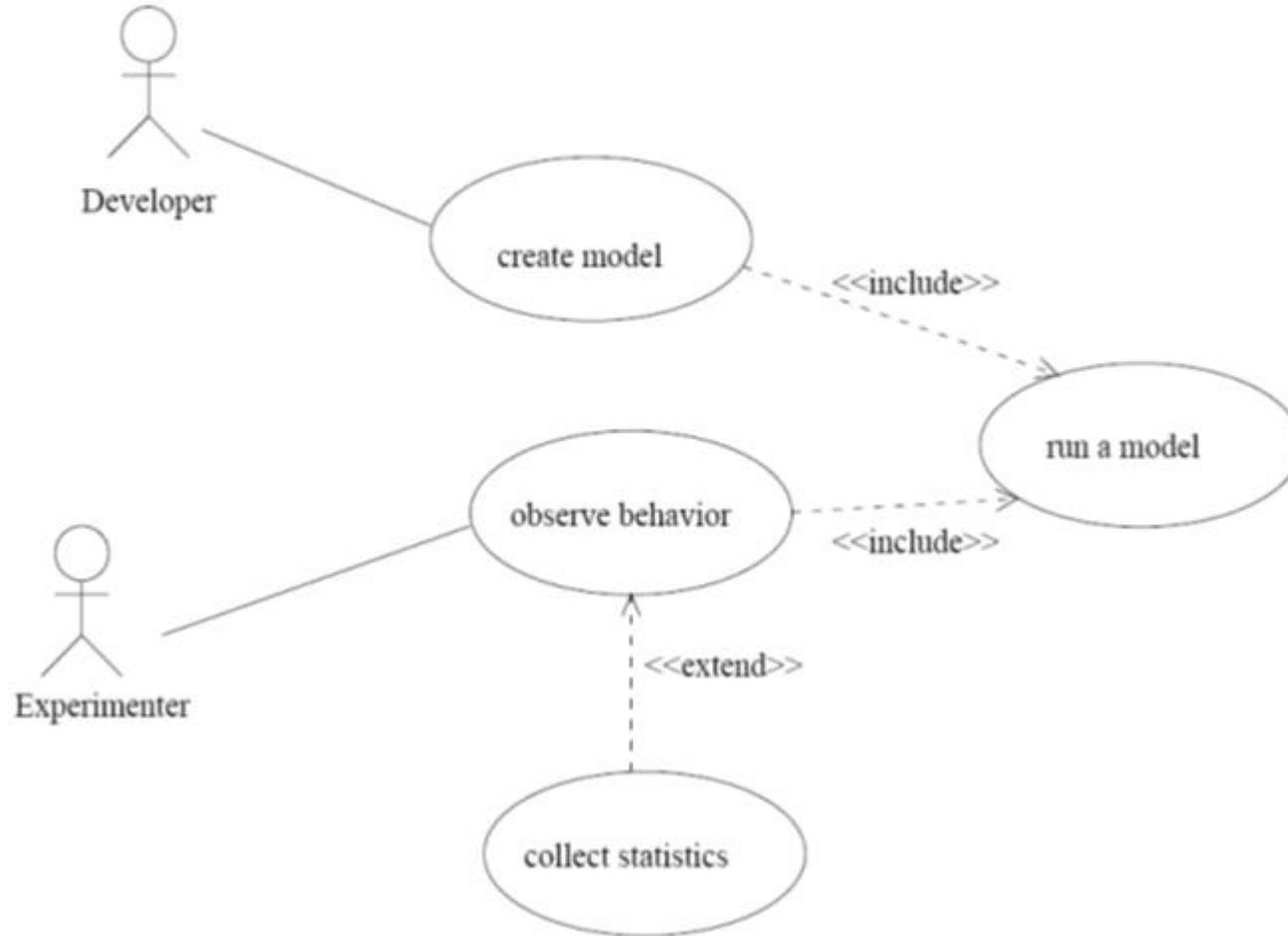


<<include>>

Refinamiento <<include>>

- Uso de <<include>>
 - Cómo el sistema puede reutilizar componentes preexistentes
 - Para mostrar una funcionalidad común entre casos de uso
 - ¡Desarrollar el hecho de que se proyecta a partir de componentes existentes!
- Nota: <<include>> y <<extend>> son estereotipos UML usados para adjuntar clasificación adicional.

Uso de estereotipos

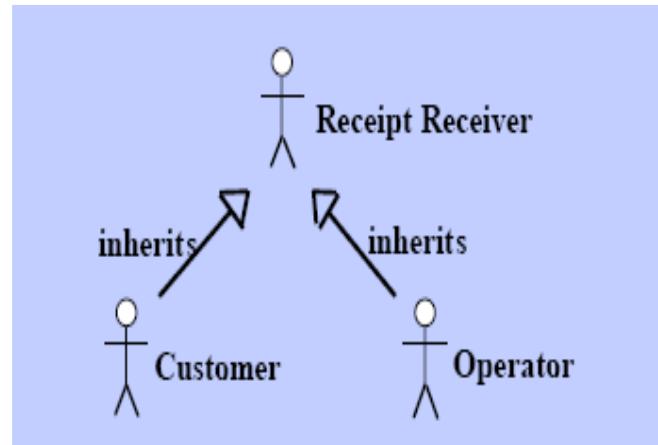
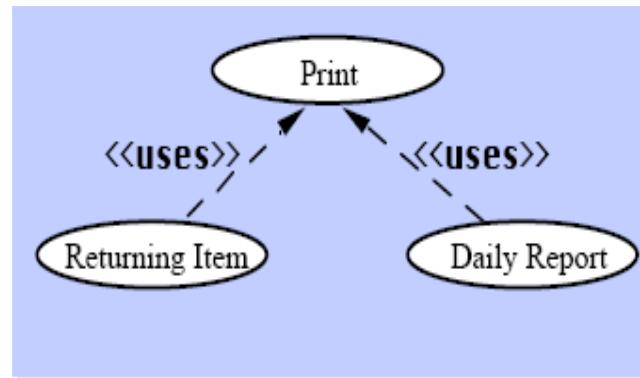
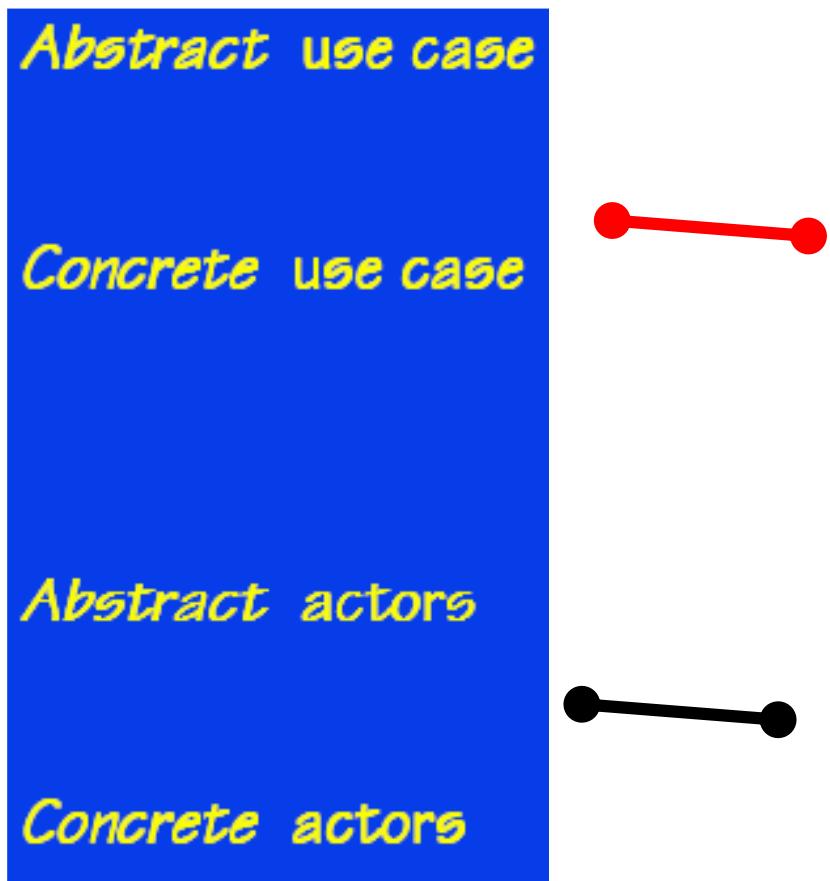


Tipo de relaciones de casos de uso

Table 5-1: Kinds of Use Case Relationships

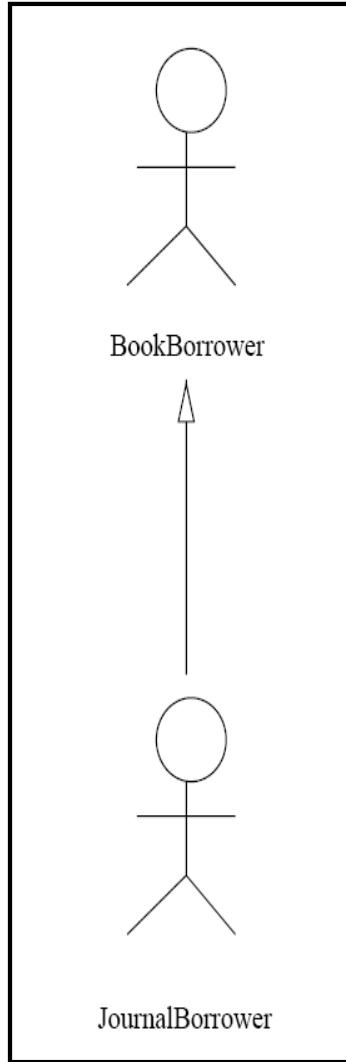
<i>Relationship</i>	<i>Function</i>	<i>Notation</i>
association	The communication path between an actor and a use case that it participates in	_____
extend	The insertion of additional behavior into a base use case that does not know about it	«extend» - - - - ⇒
use case generalization	A relationship between a general use case and a more specific use case that inherits and adds features to it	→
include	The insertion of additional behavior into a base use case that explicitly describes the insertion	«include» - - - - ⇒

Refinamiento



Generalización

Journal borrower is a book borrower



Detallando un caso de uso

- Escribir una especificación para el caso de uso
- Buenas prácticas
 - **Condiciones previas**: el estado del sistema antes de que comience el caso (es decir, hechos, cosas que deben ser ciertas)
 - **Flujo de eventos**; los pasos en el caso de uso (es decir, acciones ...)
 - **Condiciones posteriores**: el estado del sistema después de que se haya completado el caso

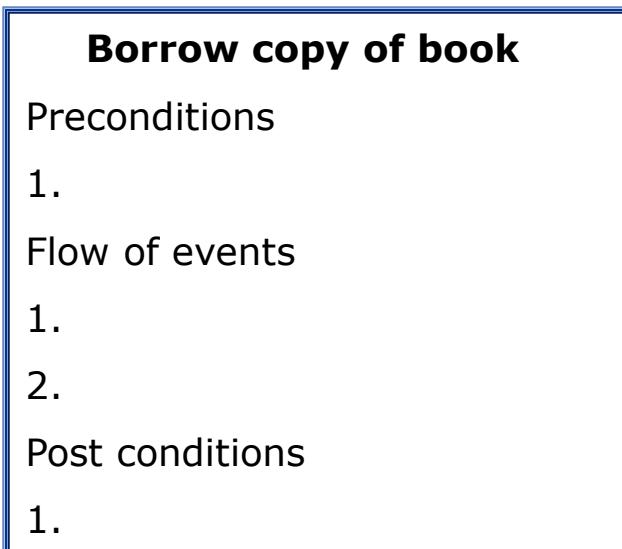
Detallando un caso de uso

- Pedir prestado una copia del libro
- Condición previa
 1. el BookBorrower es miembro de la biblioteca
 2. El BookBorrower no excede el número permitido de libros prestados
- Flujo de eventos
 1. el caso de uso comienza cuando BookBorrower intenta tomar prestado un libro
 2. el bibliotecario comprueba que está bien pedir prestado un libro
 3. Si (indique una ruta de acción alternativa)
- Condiciones posteriores
 1. el sistema ha actualizado la cantidad de libros que BookBorrower tiene en préstamo

Borrow a copy
of book

Ejercicio

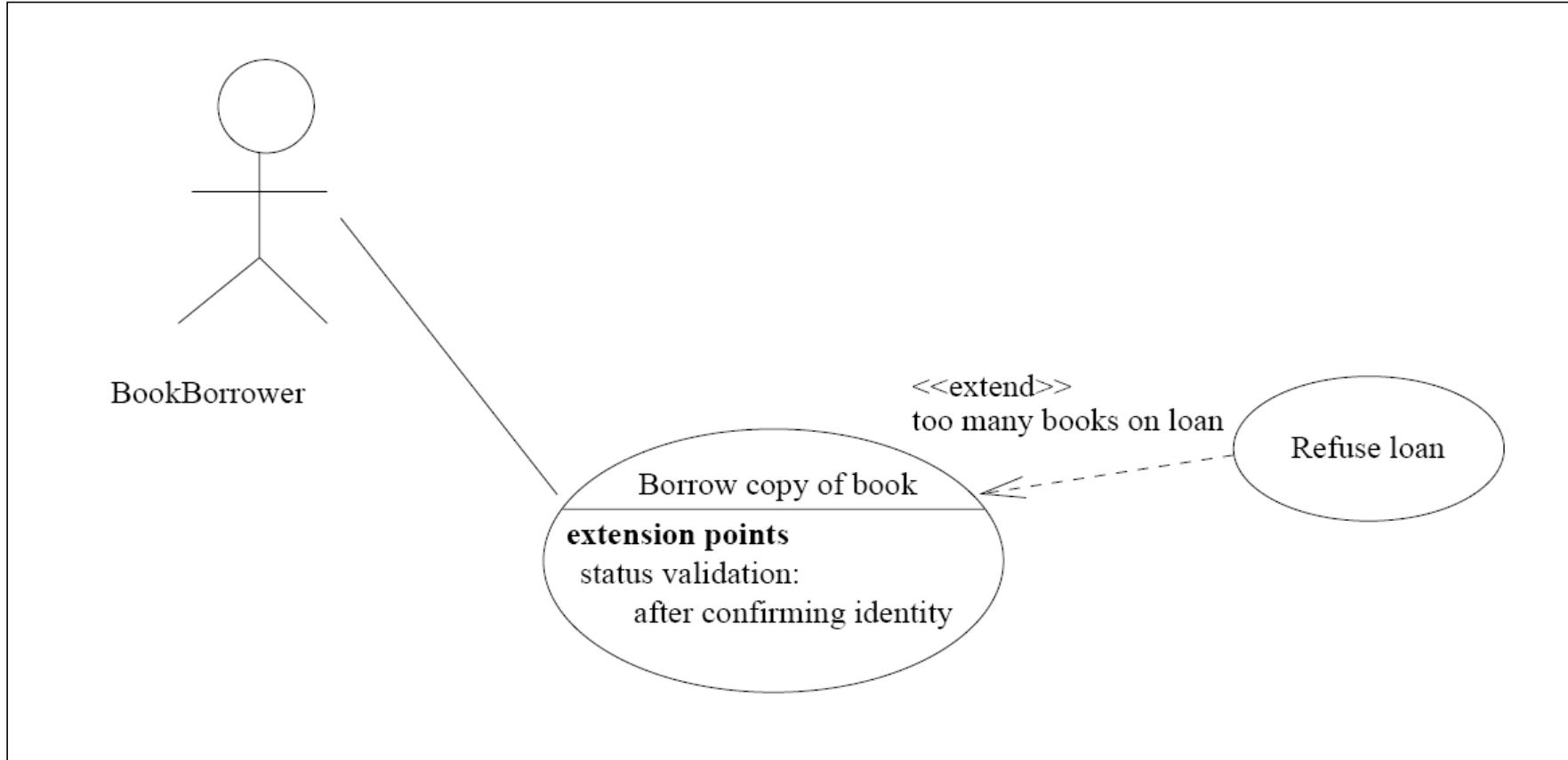
- Seleccione uno de los casos de uso identificados para el sistema de biblioteca y cree una especificación completa de él.
- Use el español estructurado para mostrar al menos un flujo alternativo de eventos y al menos una acción repetida



Escenarios

- Cada vez que un actor interactúa con un sistema, los casos de uso activados crean una instancia de un escenario
- Cada caso corresponde a una ruta específica a través de un caso de uso sin ramificación
- Los escenarios generalmente se documentan como texto junto con el caso de uso y los diagramas de actividad.

Escriba escenarios para este diagrama



Ejemplo: borrow copy of book

- Escenario 1
 - BookBorrower Joe B presta la única copia de la biblioteca de “Using UML”, cuando no tiene otro libro prestado. El sistema se actualiza consecuentemente.
- Escenario 2
 - BookBorrower Ann intenta pedir prestada la segunda copia de Diseño de Software de la biblioteca, pero se la rechazan porque tiene seis libros prestados, que es su asignación máxima.

Diagramas UML

Cubierto

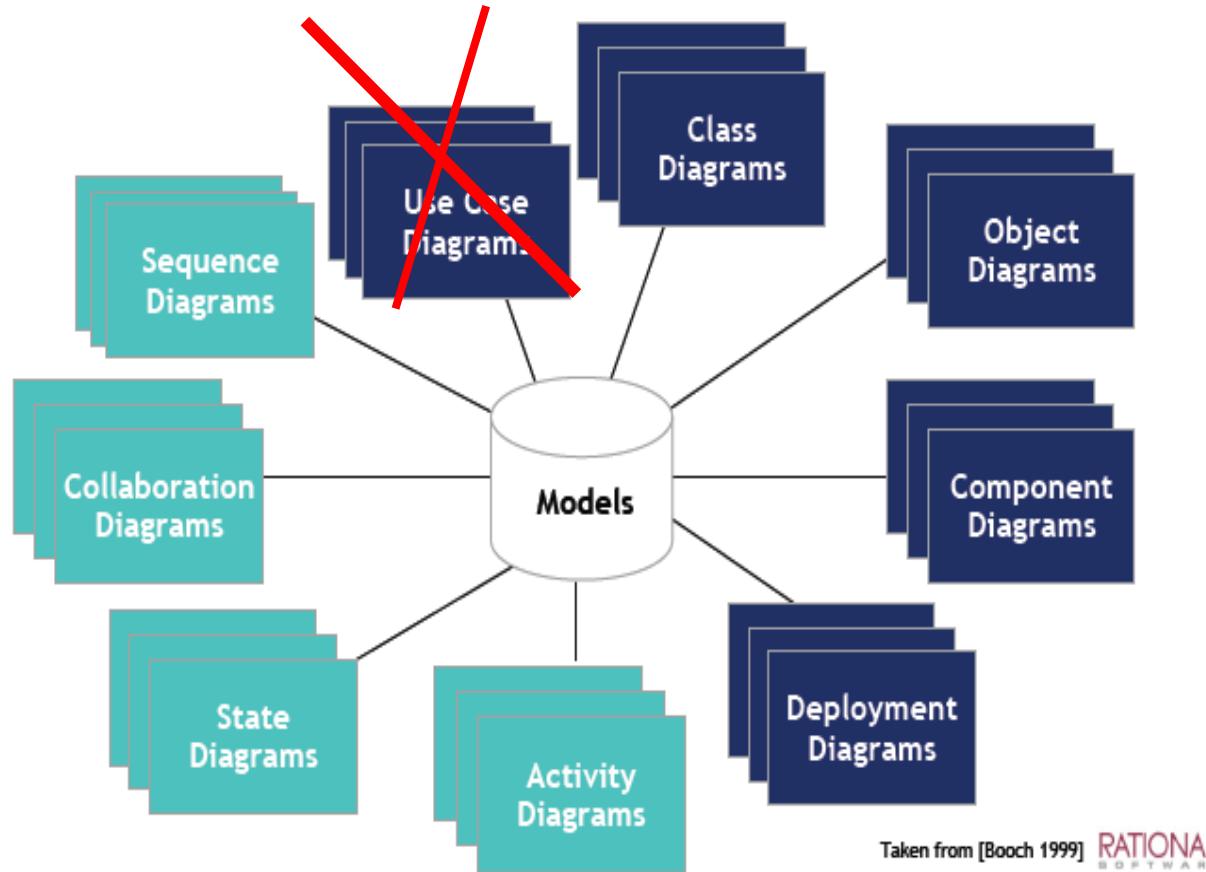
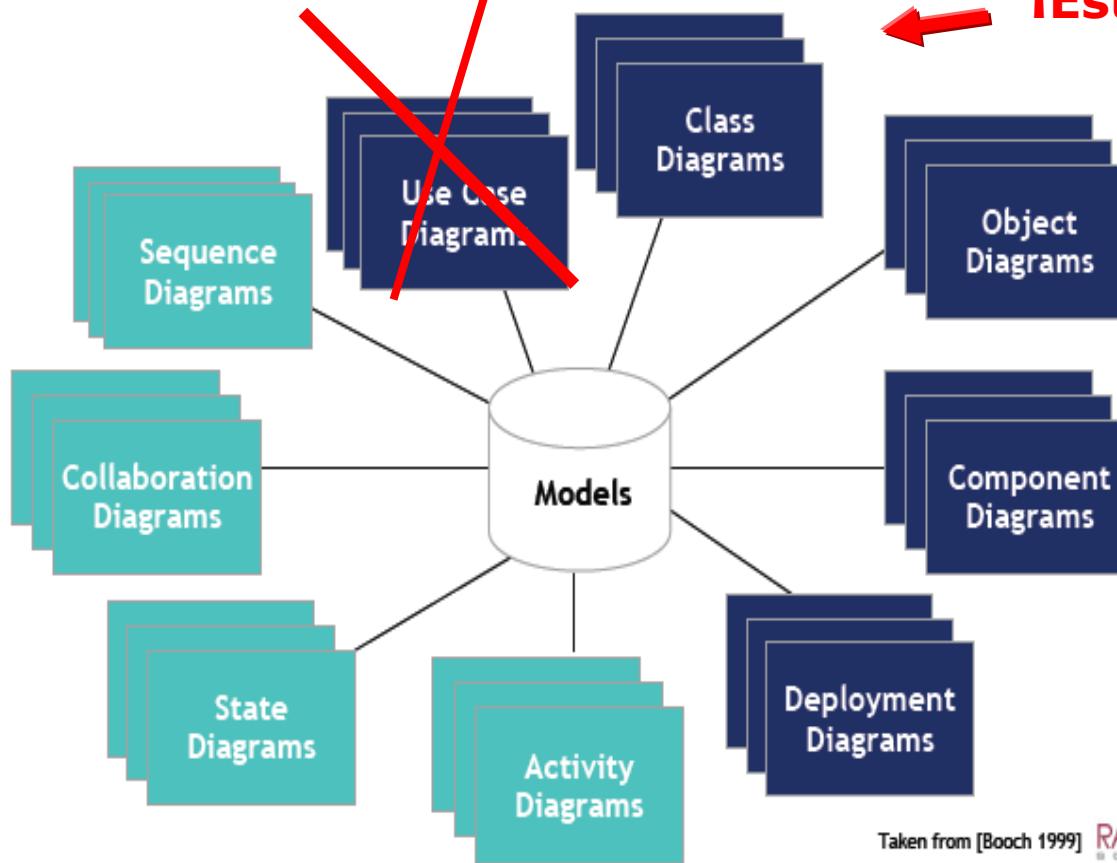


Diagrama de clases

Diagramas UML

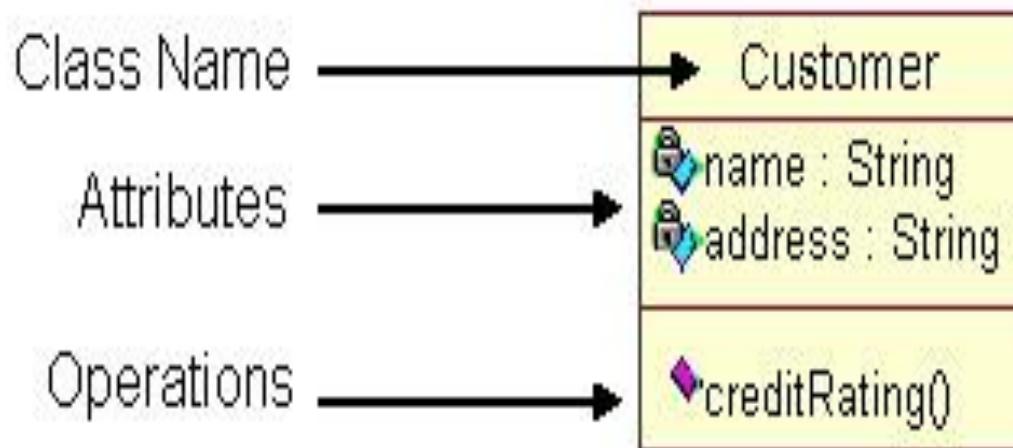
Cubierto

¡Estamos aquí!

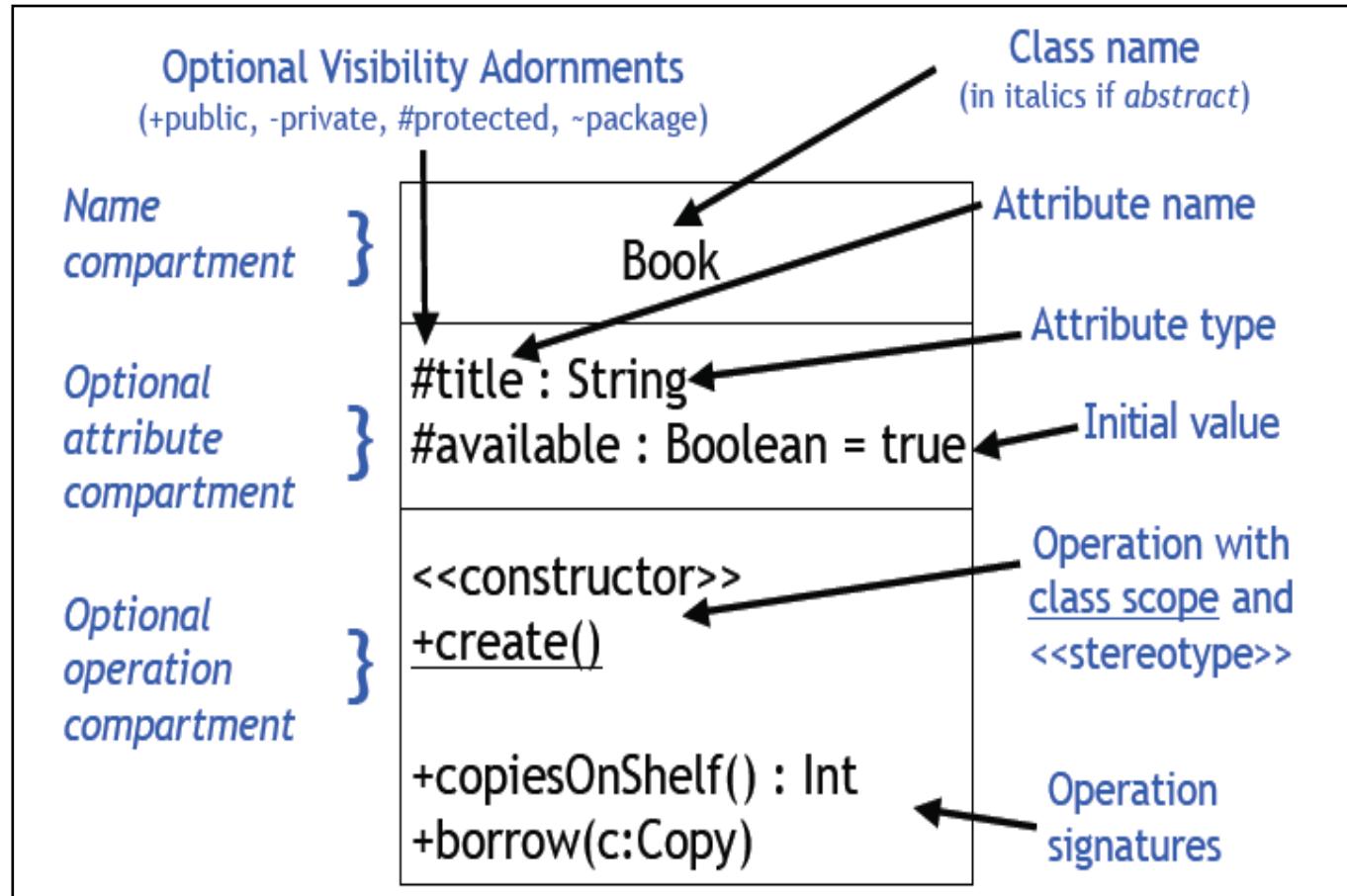


Taken from [Booch 1999] RATIONAL SOFTWARE

Clase: Ejemplo Sencillo



Elementos de una Clase UML



Reference: D. Rosenblum, UCL

+, #, -

- + significa público: cualquier cliente de la clase puede acceder a los miembros públicos
- # significa protegido: los miembros protegidos de la clase o cualquier subclase pueden acceder a los miembros protegidos
- - significa privado: los miembros privados solo pueden acceder a miembros de la misma clase

Clases de Análisis

- Una clase de análisis abstrae una o más clases y/o subsistemas en el diseño del sistema
 - Se enfoca en el manejo de requisitos funcionales
 - Define las responsabilidades (subconjuntos coherentes de comportamiento definidos por la clase)
 - Define atributos
 - Expresa relaciones en las que participa la clase

Enfoque 1: Data-Driven Design

- Identificar todos los datos en el sistema.
- Divida en clases antes de considerar las responsabilidades.
- Enfoque común: identificación de sustantivos
 - Identifique las **clases candidatas seleccionando todos los sustantivos** y frases nominales en el documento de requisitos
 - Descartar candidatos inapropiados
 - ❖ Entidades redundantes u omnipotentes
 - ❖ Entidades vagas
 - ❖ Eventos u operaciones
 - ❖ Metalenguaje
 - ❖ Entidades fuera del alcance del sistema
 - ❖ Atributos
 - **¡Los verbos y las frases verbales resaltan las operaciones candidatas!**

Enfoque 1: Data-Driven Design

- Algunas heurísticas de qué tipo de cosas son clases [Shlaer y Mellor; Booch] ...
 - Cosas tangibles o del "mundo real": libro, copia, curso;
 - Roles- miembro de la biblioteca, estudiante, director de estudios,
 - Eventos- llegada, salida, solicitud;
 - Interacciones- reunión, intersección

Ejercicio

- Realice un análisis sustantivo-verbo de su documento de requerimientos;
- Subraye todos los sustantivos y frases nominales,
- Cree una lista de clases candidatas (al examinar los criterios de descarte, también puede identificar algunos atributos candidatos)
- Identifique todos los verbos y frases verbales
- Cree una lista de operaciones candidatas y asignarlas a clases.

Ejercicio

- **Books and journals** The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Only members of staff may borrow journals. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. New books and journals arrive regularly, and old ones are sometimes disposed of. The current year's journals are sent away to be bound into volumes at the end of each year.
- **Borrowing** It is essential that the system keep track of when books and journals are borrowed and returned, since the current system already does that. The new system should produce reminders when a book is overdue. There may in future be a requirement for users to be able to extend the loan of a book if it is not reserved.
- **Browsing** The system should allow users to search for a book on a particular topic, by a particular author, etc. to check whether a copy of the book is available for loan and, if not, to reserve the book. Anybody can browse in the library.

Análisis sustantivo/verbo

Books and journals The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.

Borrowing The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

Enfoque 2: Responsibility-Driven Design

- Identificar todas las responsabilidades en el sistema.
- Divide en clases antes de considerar los datos de las clases.
- Enfoque común: tarjetas CRC
 - Clase, Responsabilidades, Colaboraciones

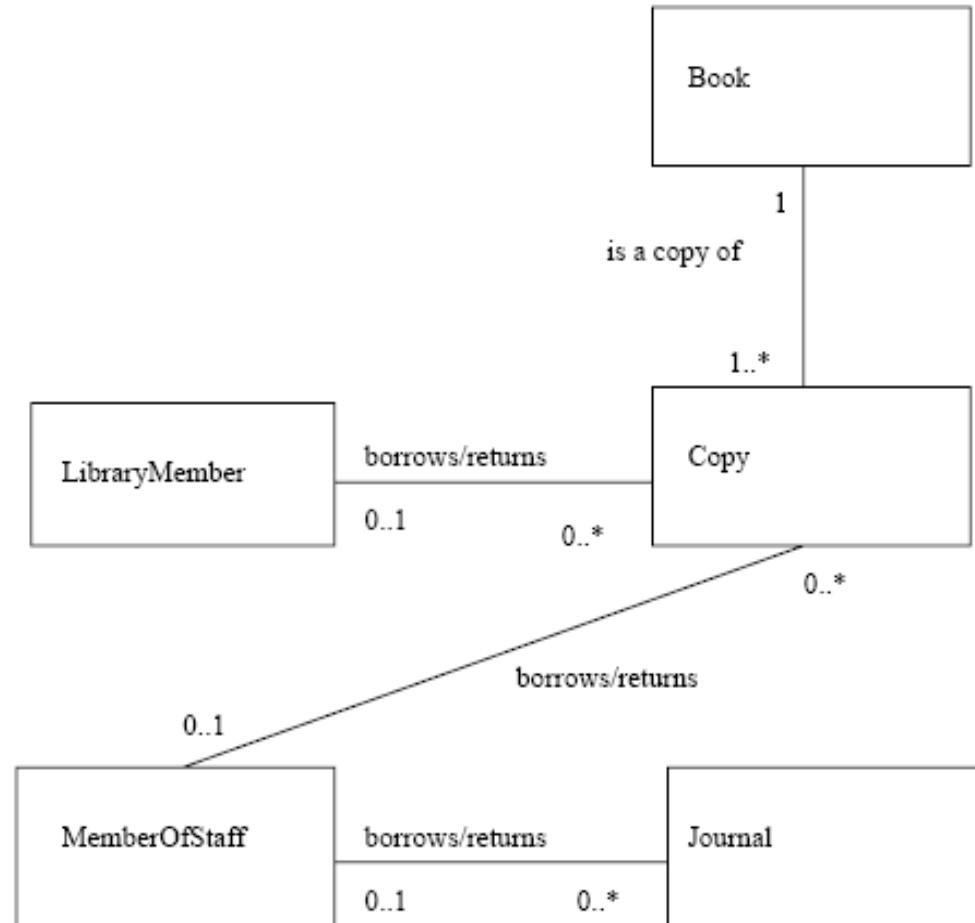
Ejemplo de tarjetas CRC para una biblioteca

LibraryMember	
Responsibilities	Collaborators
Maintain data about copies currently borrowed	
Meet requests to borrow & return copies	Copy
Copy	
Responsibilities	Collaborators
Maintain data about particular copy of book	
Inform corresponding Book when borrowed/returned	Book
Book	
Responsibilities	Collaborators
Maintain data about one book	
Know whether there are borrowable copies	

Ejercicio

- Realice un análisis basado en la responsabilidad del sistema para identificar posibles clases:
 - Mire el documento de requerimientos y los casos de uso
 - Identifique clases candidatas.
- Derive su CRC (es decir, clase, responsabilidad y colaboradores)

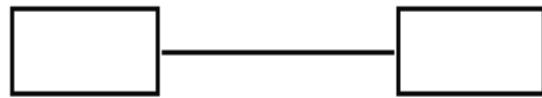
First-Cut de Diagrama de Clases



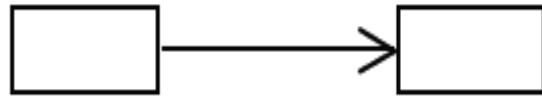
Relaciones

- Las relaciones son conexiones entre elementos de modelado.
- Mejoran la comprensión del dominio, describiendo cómo los objetos trabajan juntos
- Actúan como un control de consistencia para un buen modelado
- **Asociaciones** son relaciones entre clases.
- Ejemplos
 - El objeto de la clase A envía un mensaje al objeto de la clase B
 - El objeto de la clase A crea un objeto de la clase B
 - El objeto de la clase A tiene un atributo cuyos valores son objetos de la clase B
 - El objeto de la clase A recibe un mensaje con argumento de la clase B
- **Enlaces** son relaciones entre objetos
 - Los enlaces pueden ser instancias de asociaciones (como en UML 1.4)
 - Permiten que un objeto invoque operaciones en otro objeto

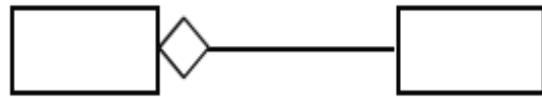
Notación de Relaciones en UML



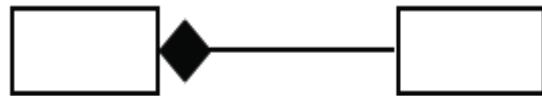
bidirectional / binary



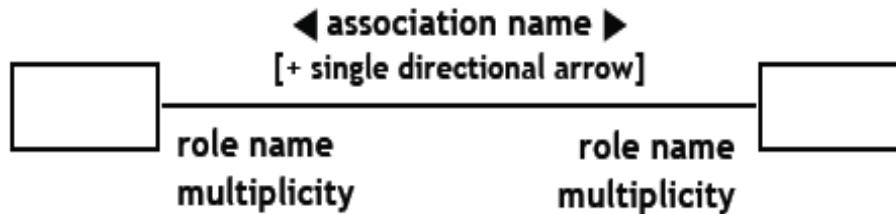
unidirectional



aggregation



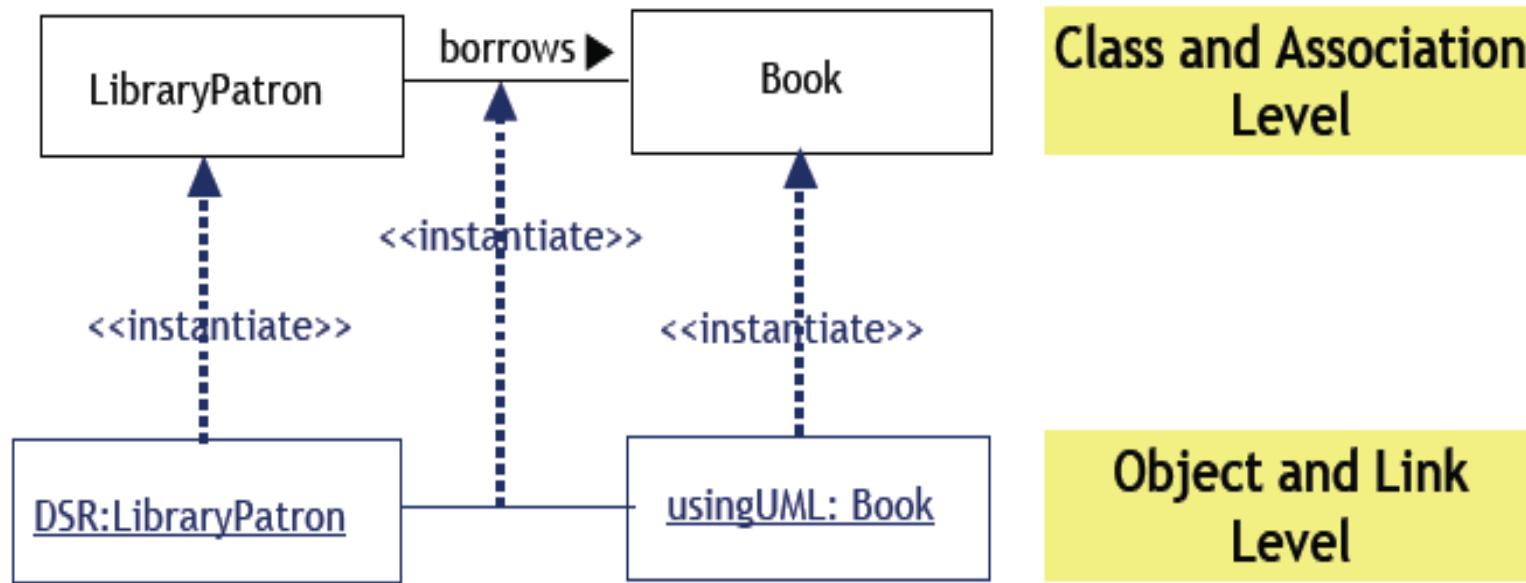
composition



supplementary
characteristics

Reference: D. Rosenblum, UCL

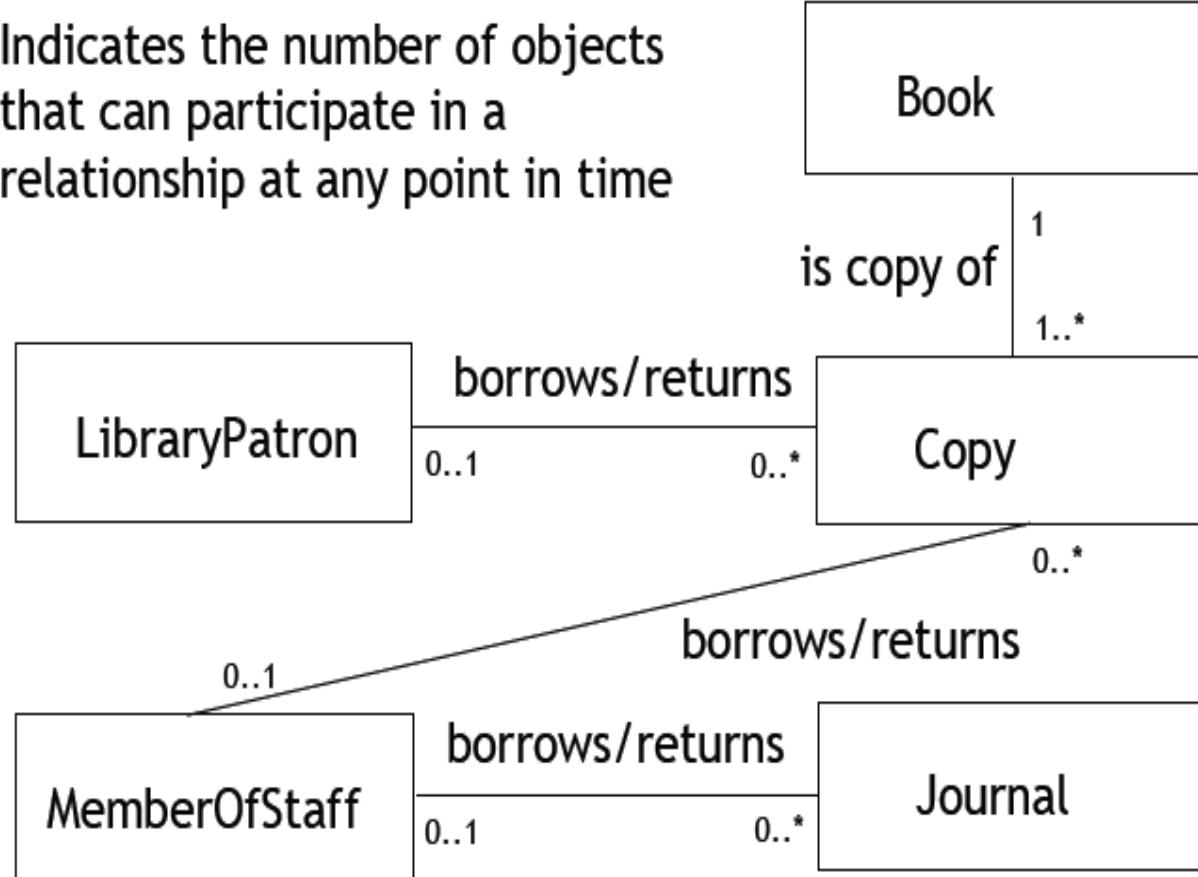
Los Enlaces instancian Asociaciones



Reference: D. Rosenblum, UCL

Multiplicidad de una Asociación

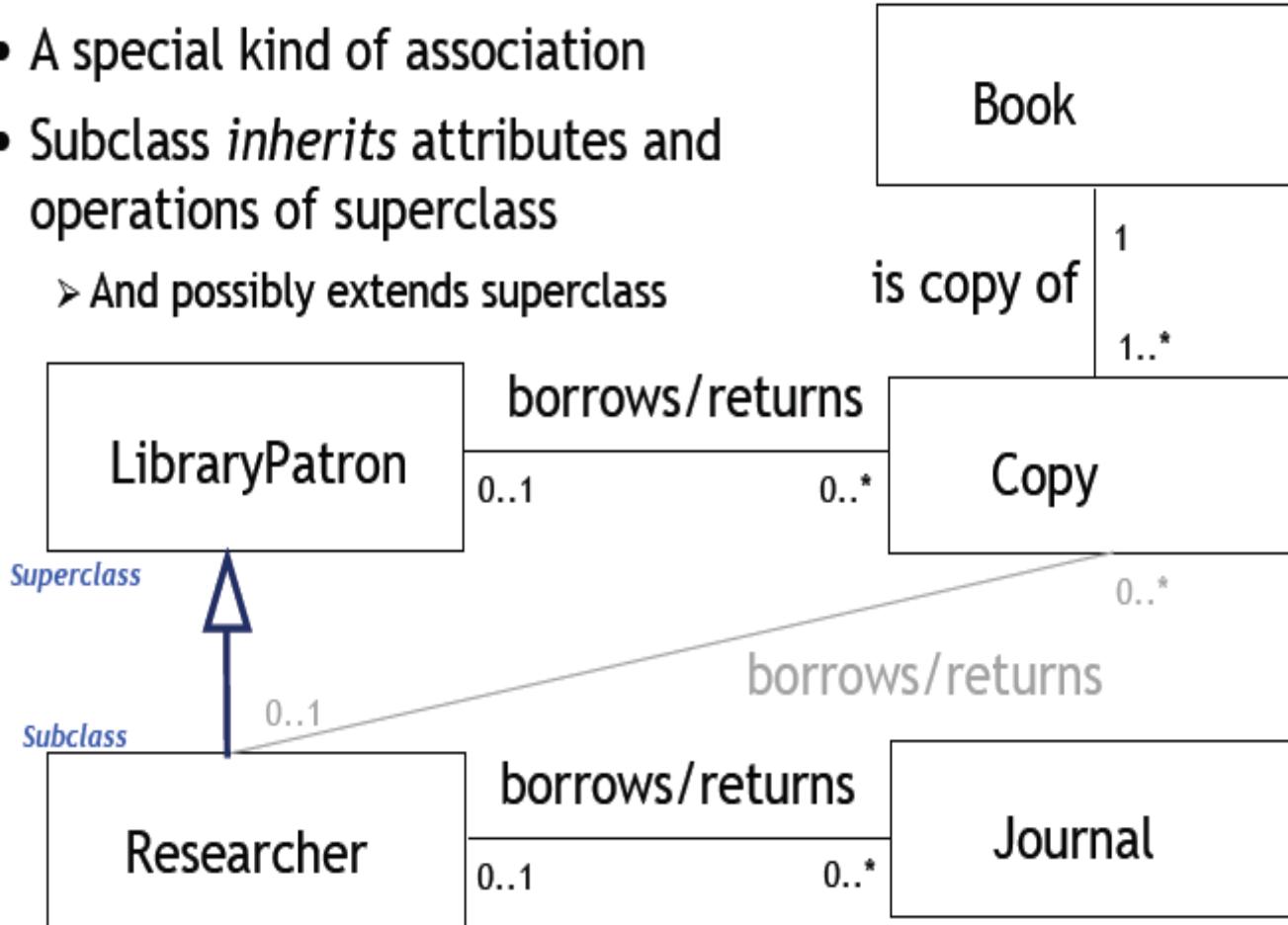
- Indicates the number of objects that can participate in a relationship at any point in time



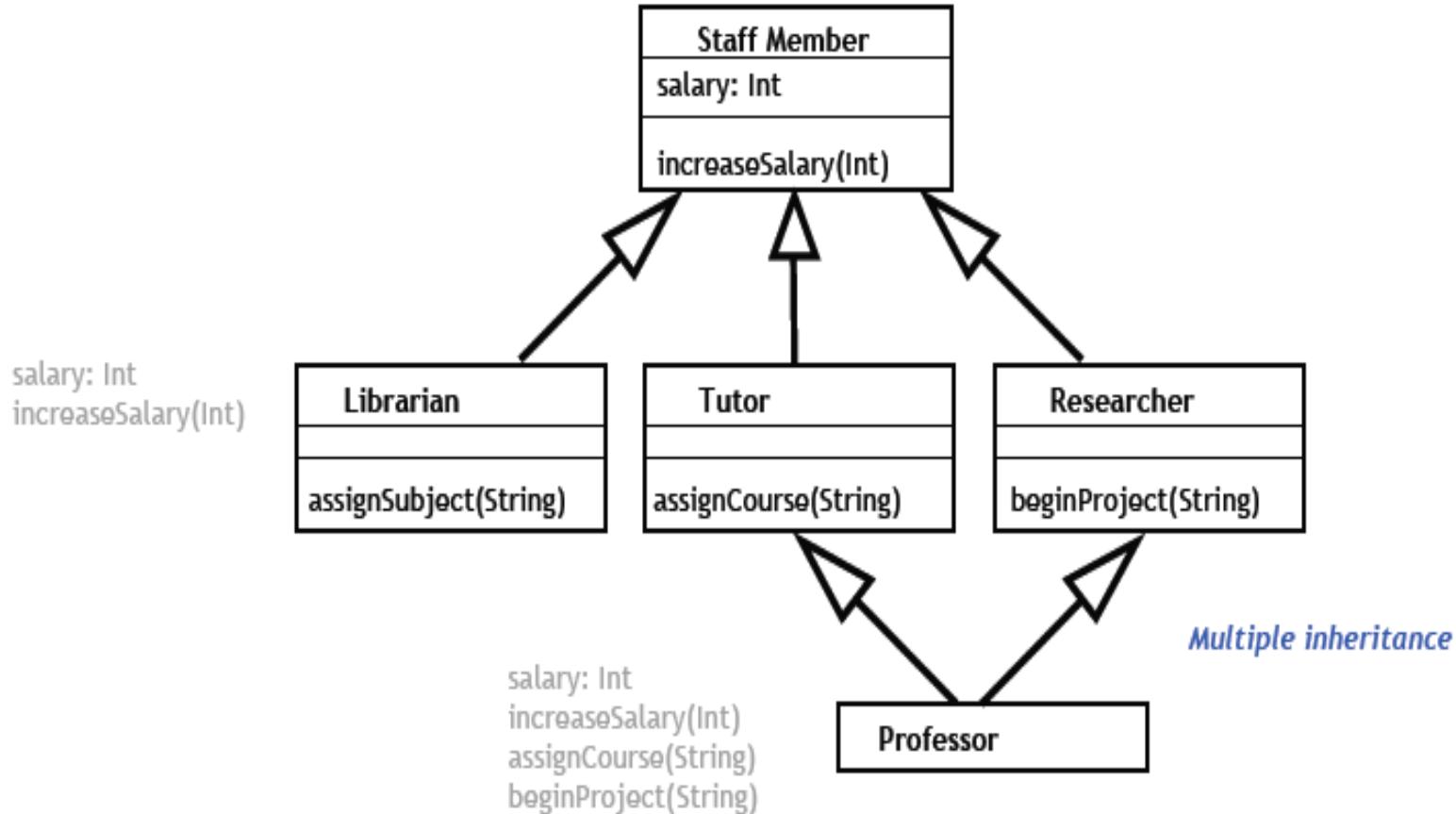
Reference: D. Rosenblum, UCL

Generalización y Herencia

- A special kind of association
- Subclass *inherits* attributes and operations of superclass
 - And possibly extends superclass



Otro Ejemplo de Herencia



Asociaciones Parte/Todo

- **Aggregation:** Weak Ownership

- The part objects can feature simultaneously in any number of other whole objects

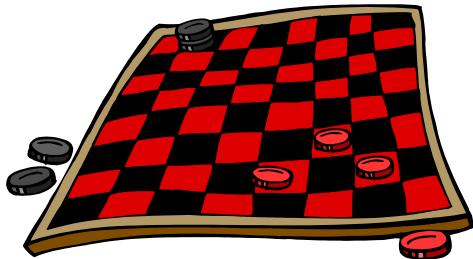


A module is part of a course

In fact,

5 or more modules are part of one or more courses

Part/Whole Associations



Compuesto de 64 cuadrados

- **Composition:** Strong Ownership
 - The whole strongly owns its parts, so the parts cannot feature elsewhere



- **NOTE:** Not all 1-to-* relationships imply ownership

Clases de Asociación

- Se usa para adjuntar atributos a una asociación en lugar de las clases que asocia
- ¡La línea de asociación de clase debe tener el mismo nombre!

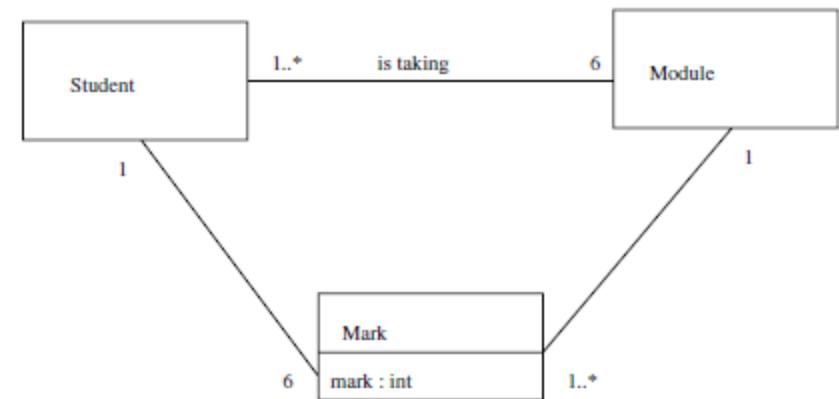
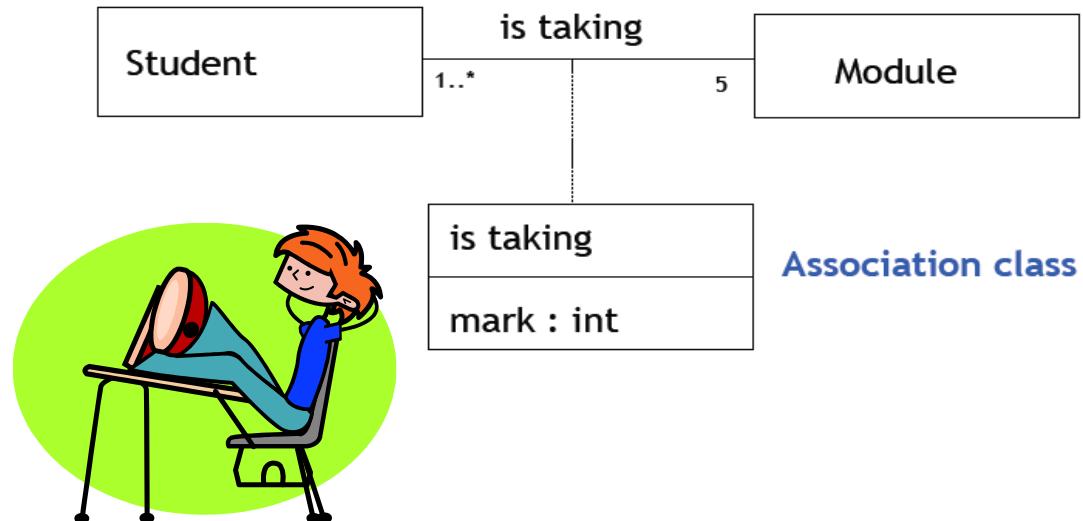
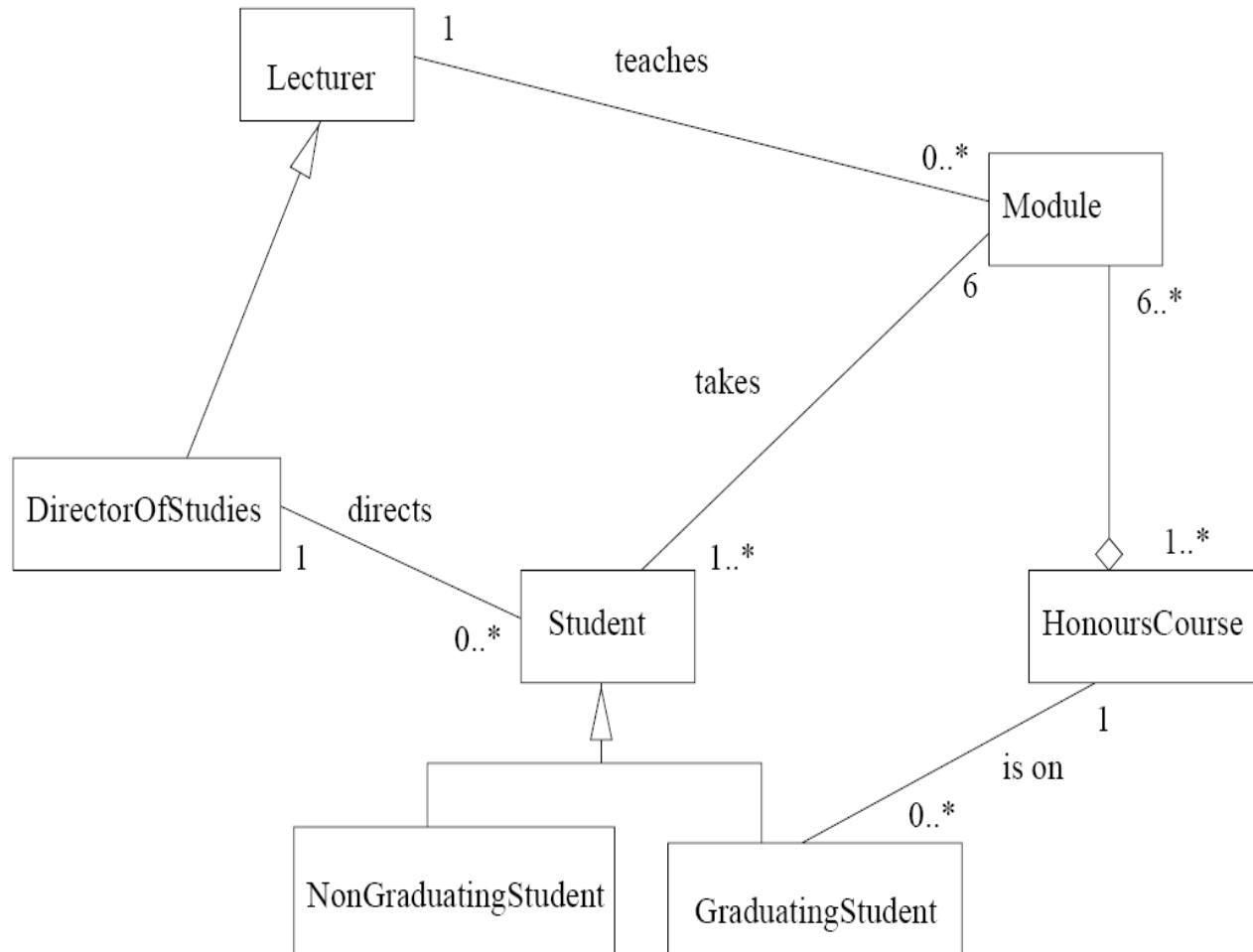
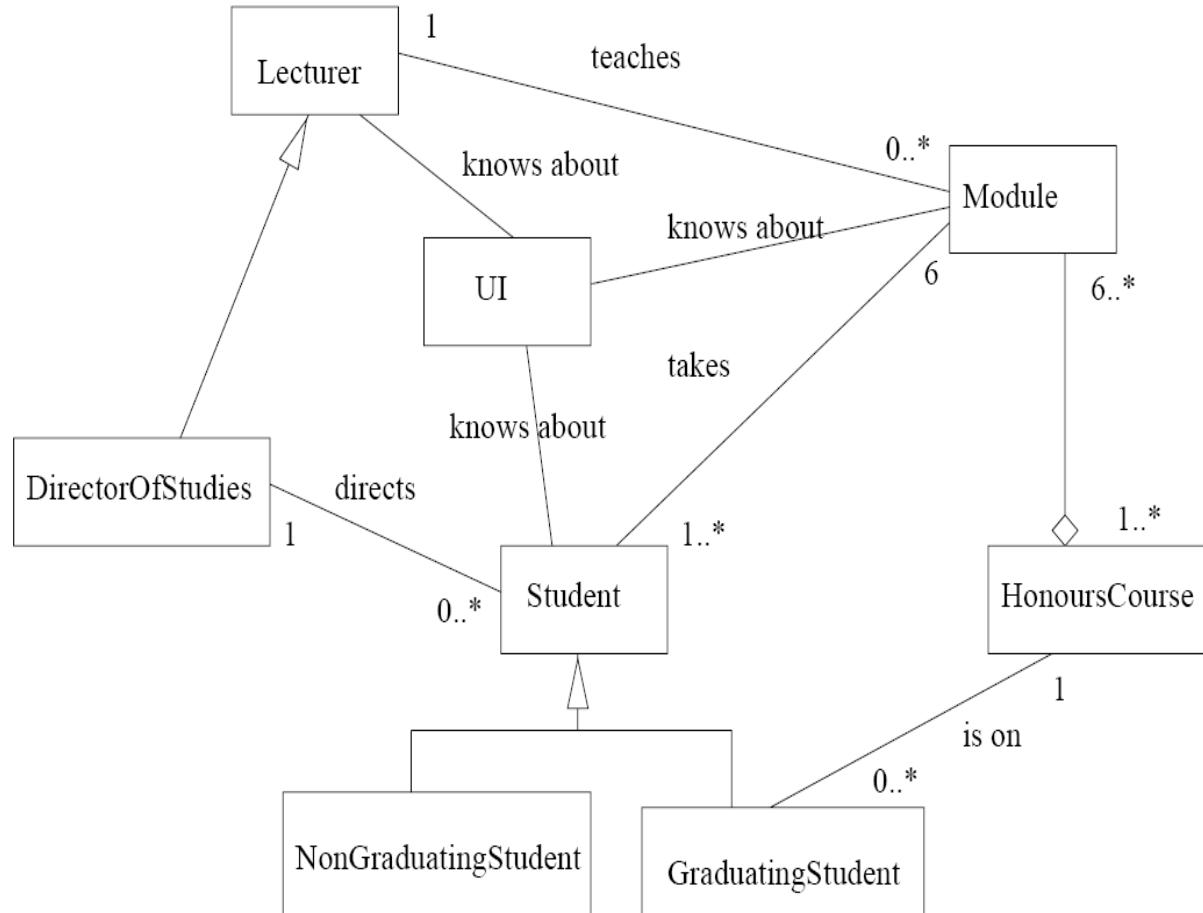


Figure 6.13 Avoiding an association class.

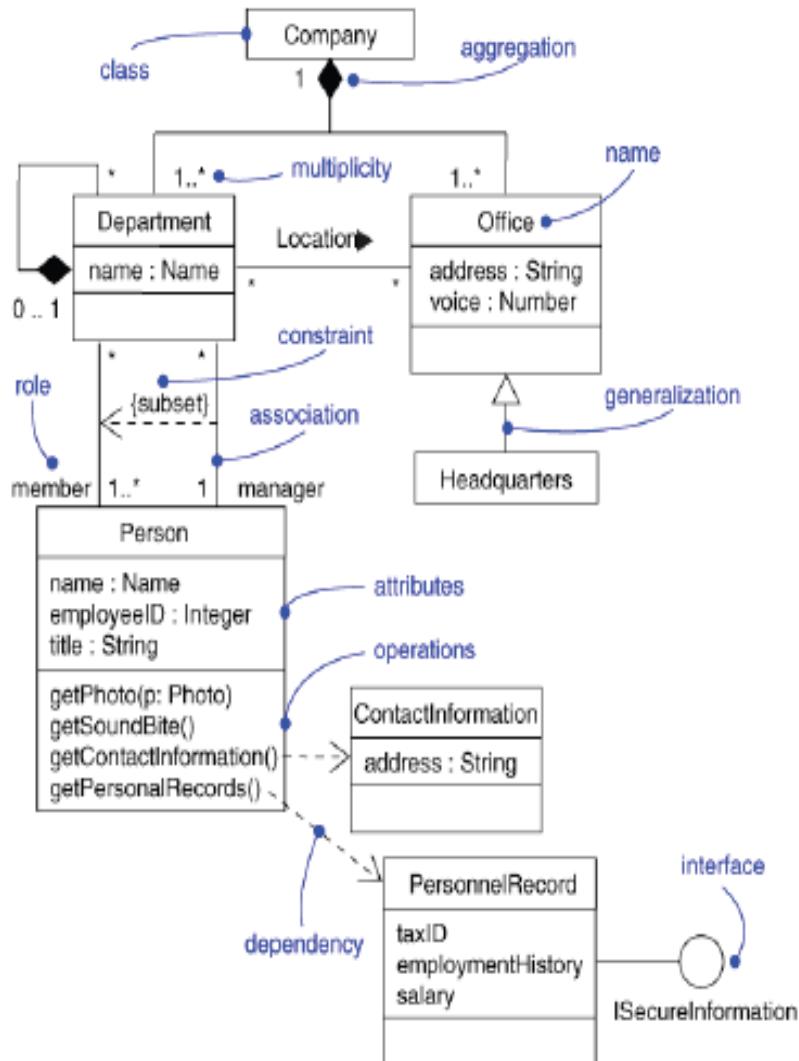
Ejemplo: Modelo de Clases



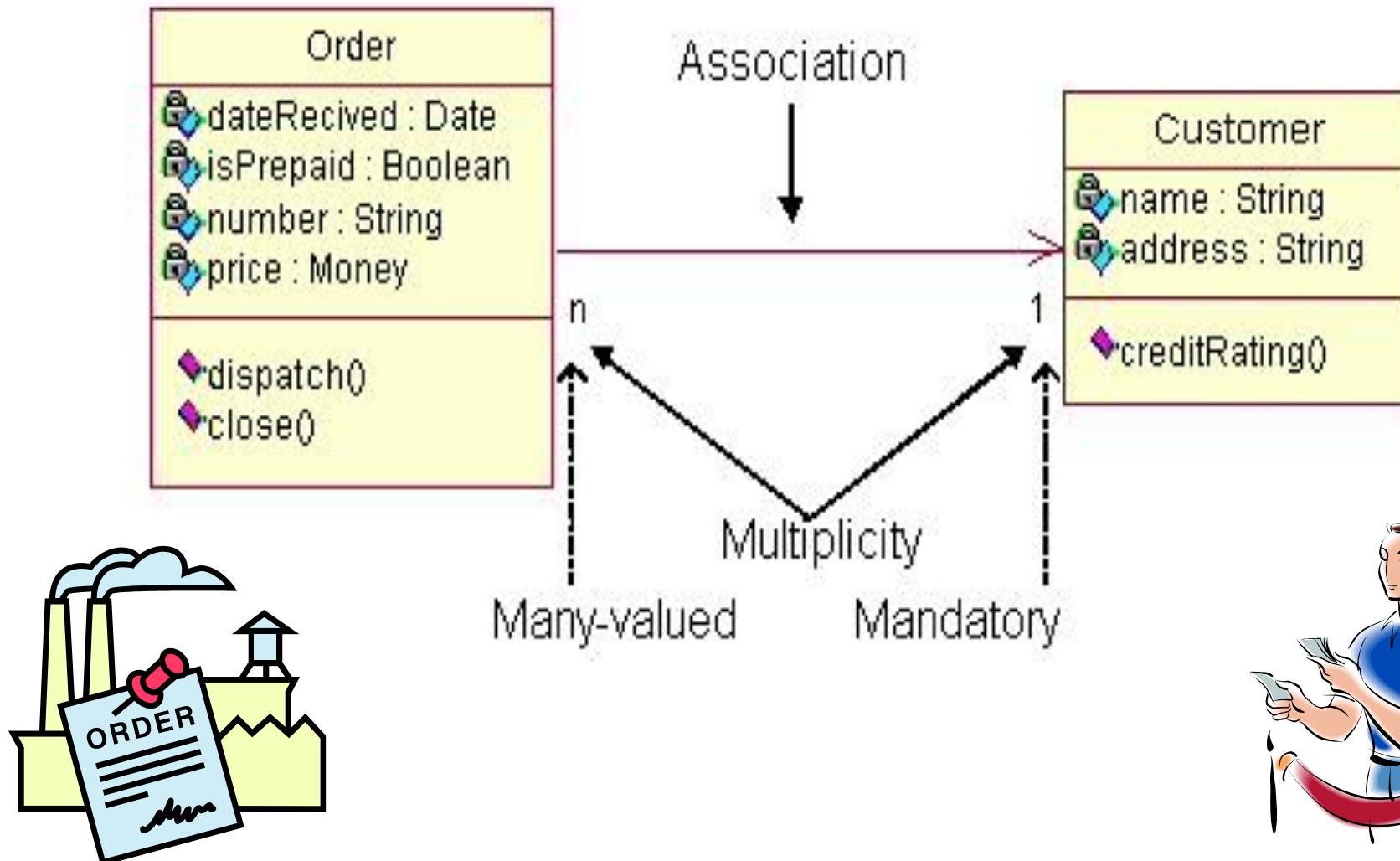
Otro Ejemplo: Modelo de Clases



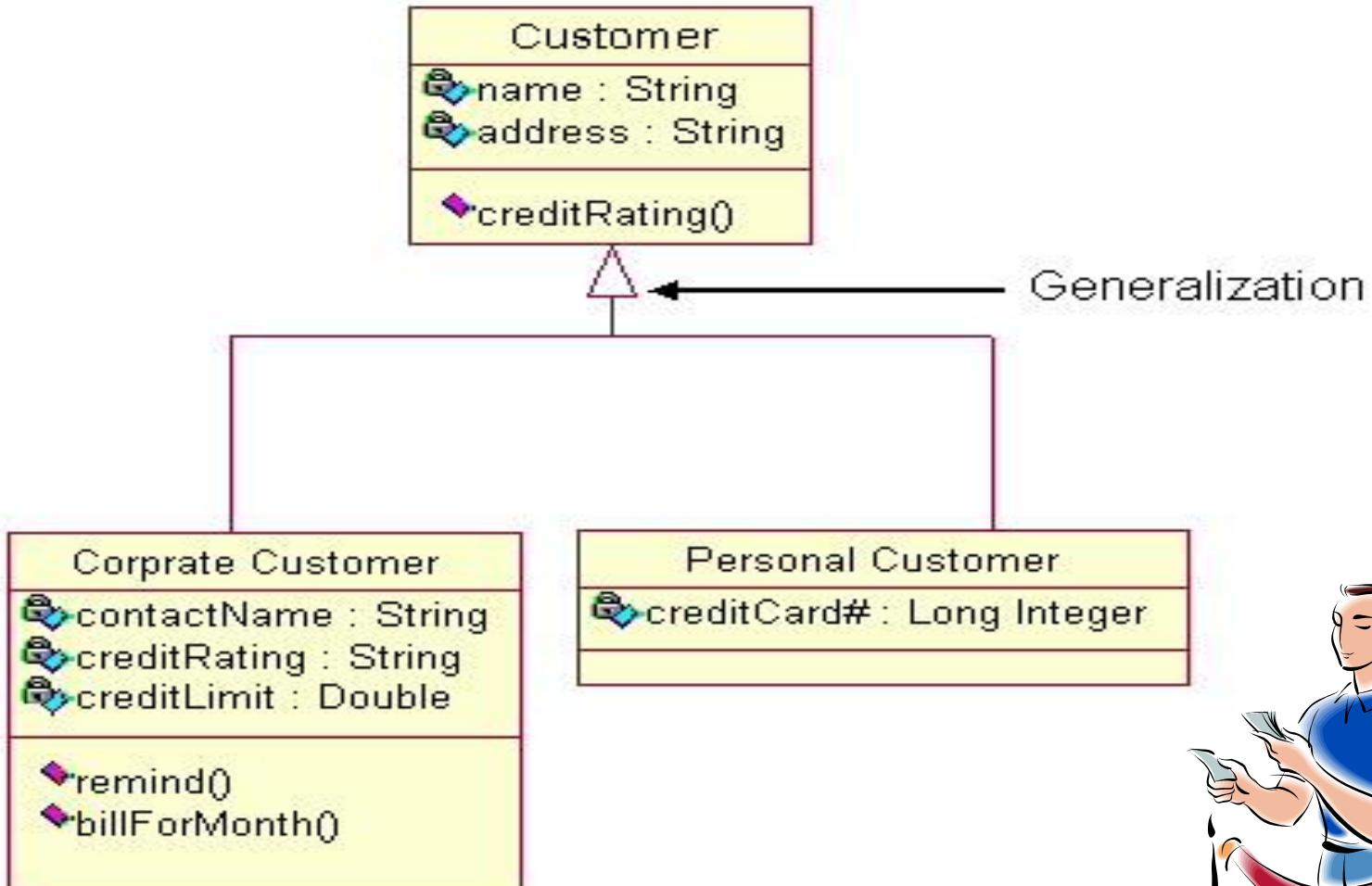
Ejemplo: Diagrama de Clases



More Examples

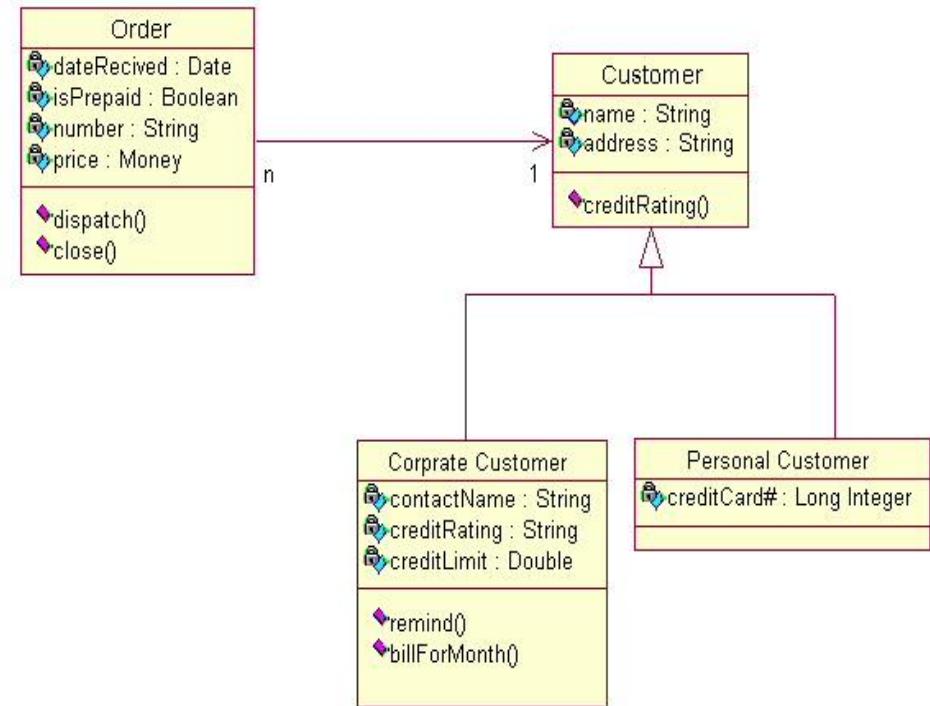


Más ejemplos



Más Ejemplos

- Las clases **Cliente Corporativo** y **Cliente Personal** tienen algunas similitudes, como el nombre y la dirección, pero cada clase tiene algunos de sus propios atributos y operaciones.
- La clase **Cliente** es una forma general tanto del Cliente corporativo y clases de clientes personales.



¿Qué hace “Buena” a una Clase de Análisis...?

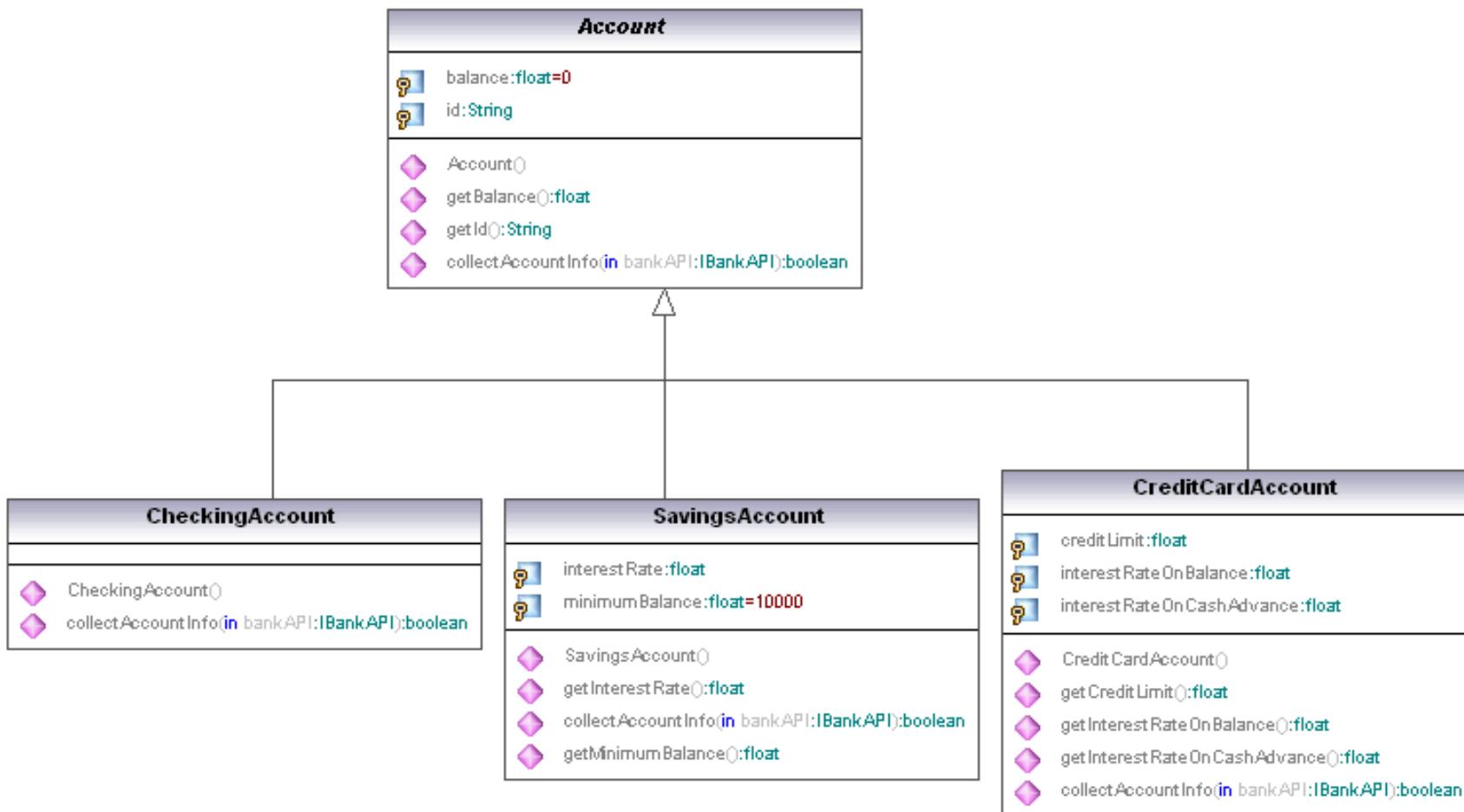
- Su nombre refleja su intención.
- Es una abstracción nítida que modela un elemento específico del dominio del problema.
- Tiene un conjunto pequeño pero definido de responsabilidades.
- Tiene **alta cohesión**
- Tiene **bajo acoplamiento** con otras clases.
- Tarea:
 - ¿Cómo se relaciona la abstracción con la cohesión?
 - ¿Cómo se relaciona el encapsulamiento con el acoplamiento?



Nota...

- Análisis de sustantivo/verbo y análisis basado en la responsabilidad
 - Sustantivo / verbo y responsabilidad se complementan entre sí
 - A menudo va de la mano con casos de uso
- Diagrama de clase de first-cut (también referido al modelo de clases)
- Refina el diagrama de first-cut en un diagrama de clases detallado

Pista...



Diagramas UML

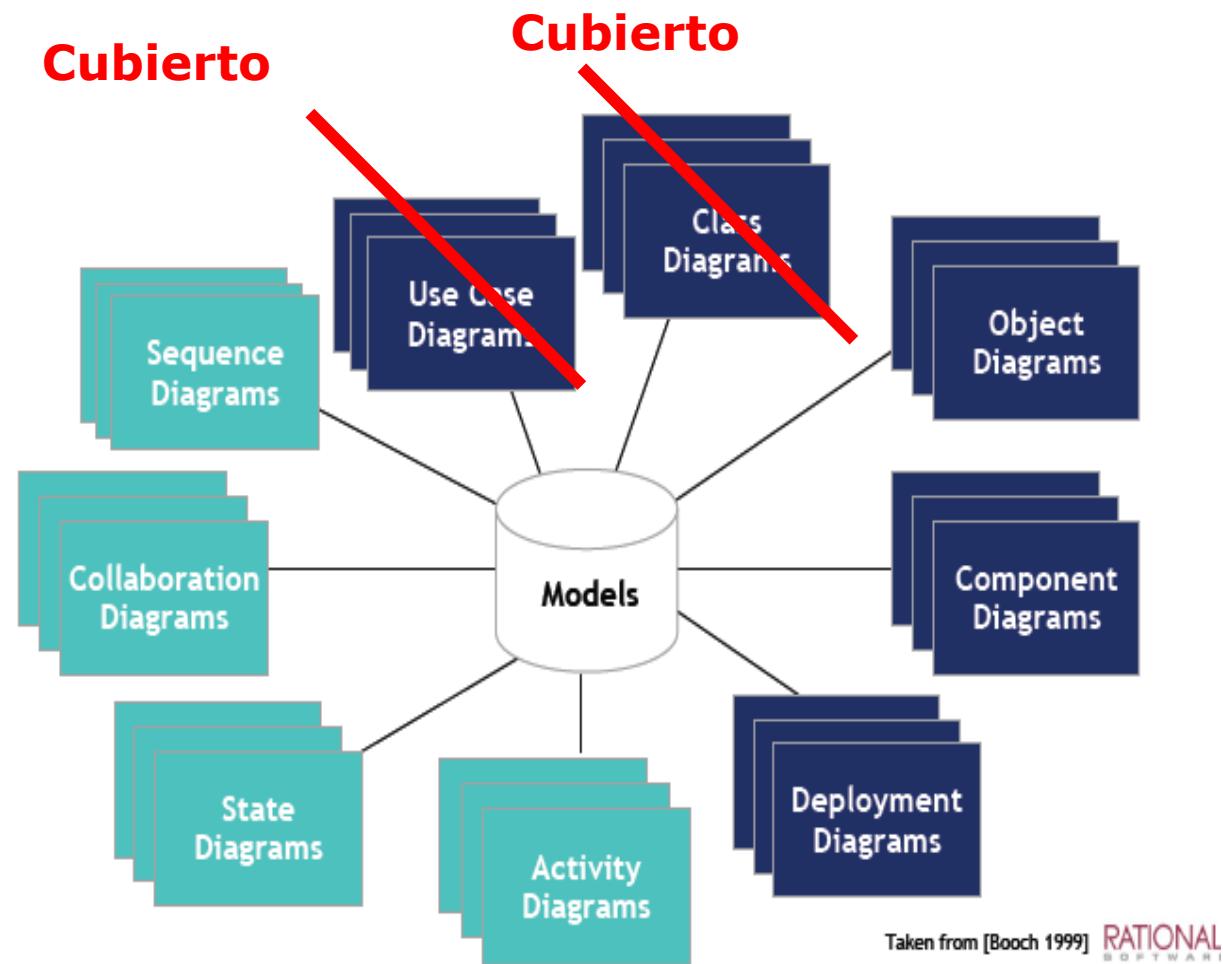


Diagrama de secuencia

Diagramas UML

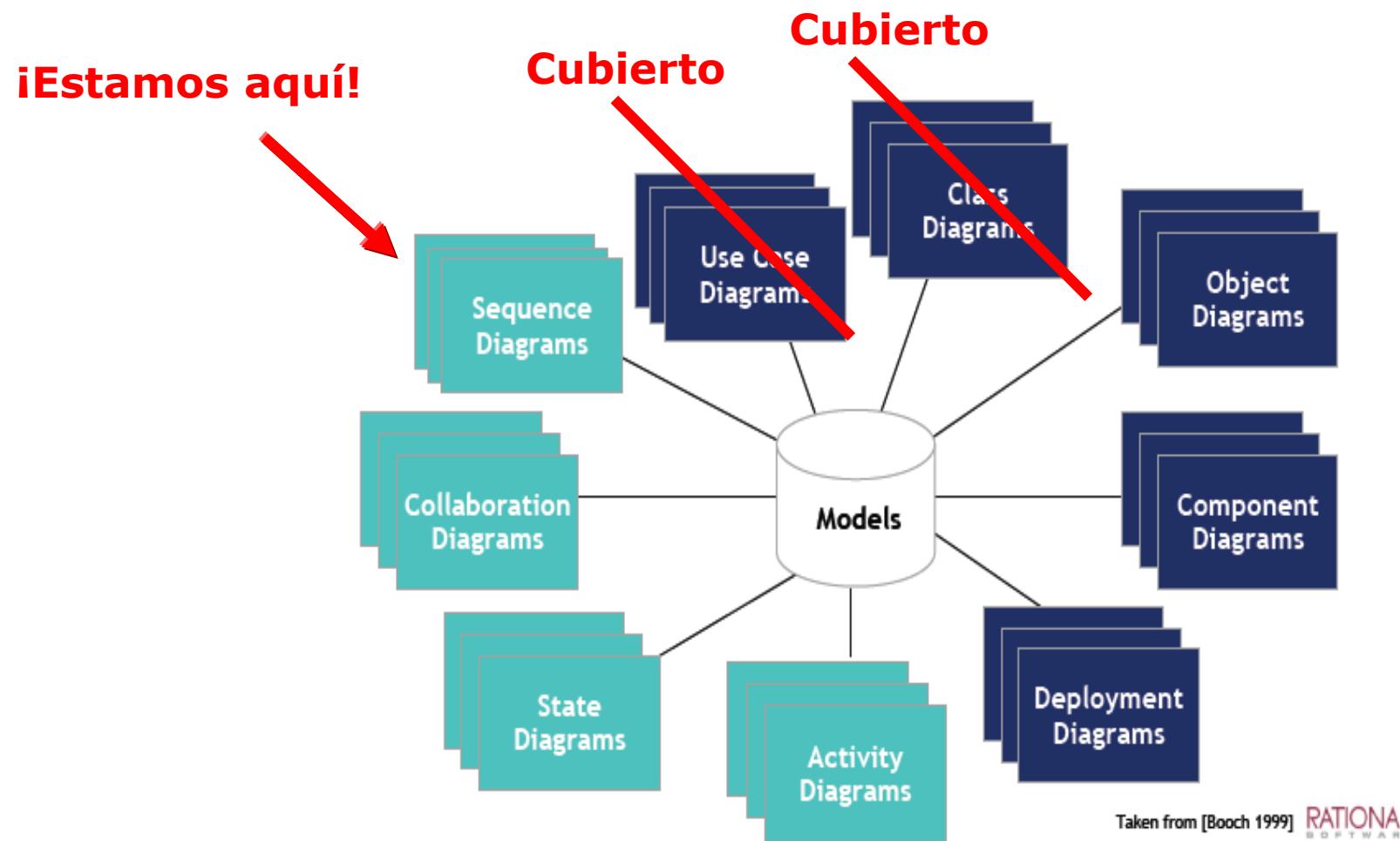


Diagrama de Secuencia

- Los diagramas de secuencia demuestran el **comportamiento** de los objetos en un caso de uso al describir los objetos y los mensajes que transmiten. los diagramas se **leen de izquierda a derecha y en forma descendente**.
- Interacciones de objetos organizadas en una **secuencia de tiempo** (es decir, **orientada al tiempo**)

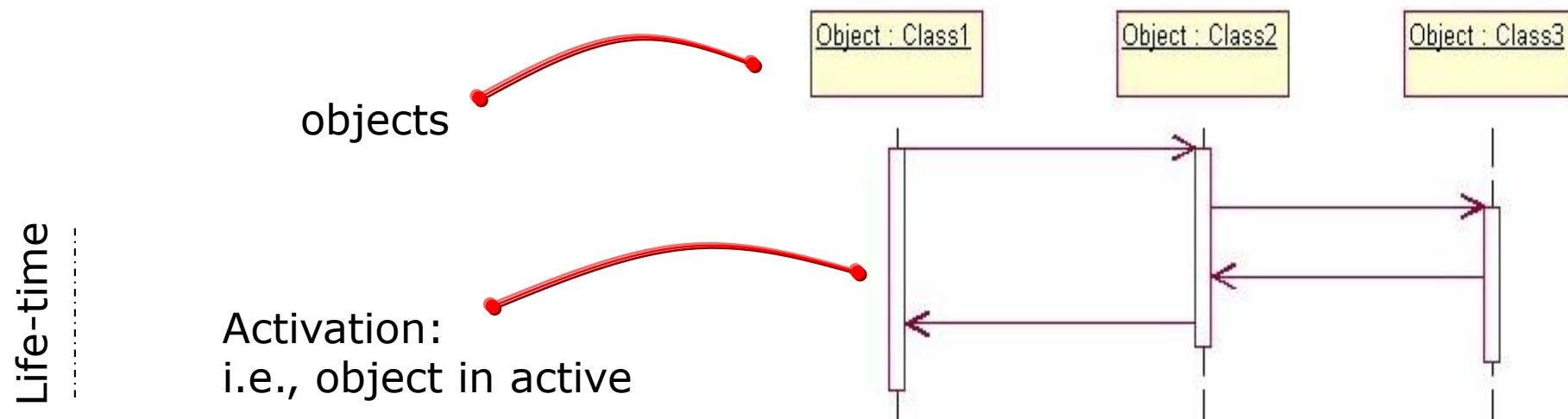


Diagrama de Secuencia

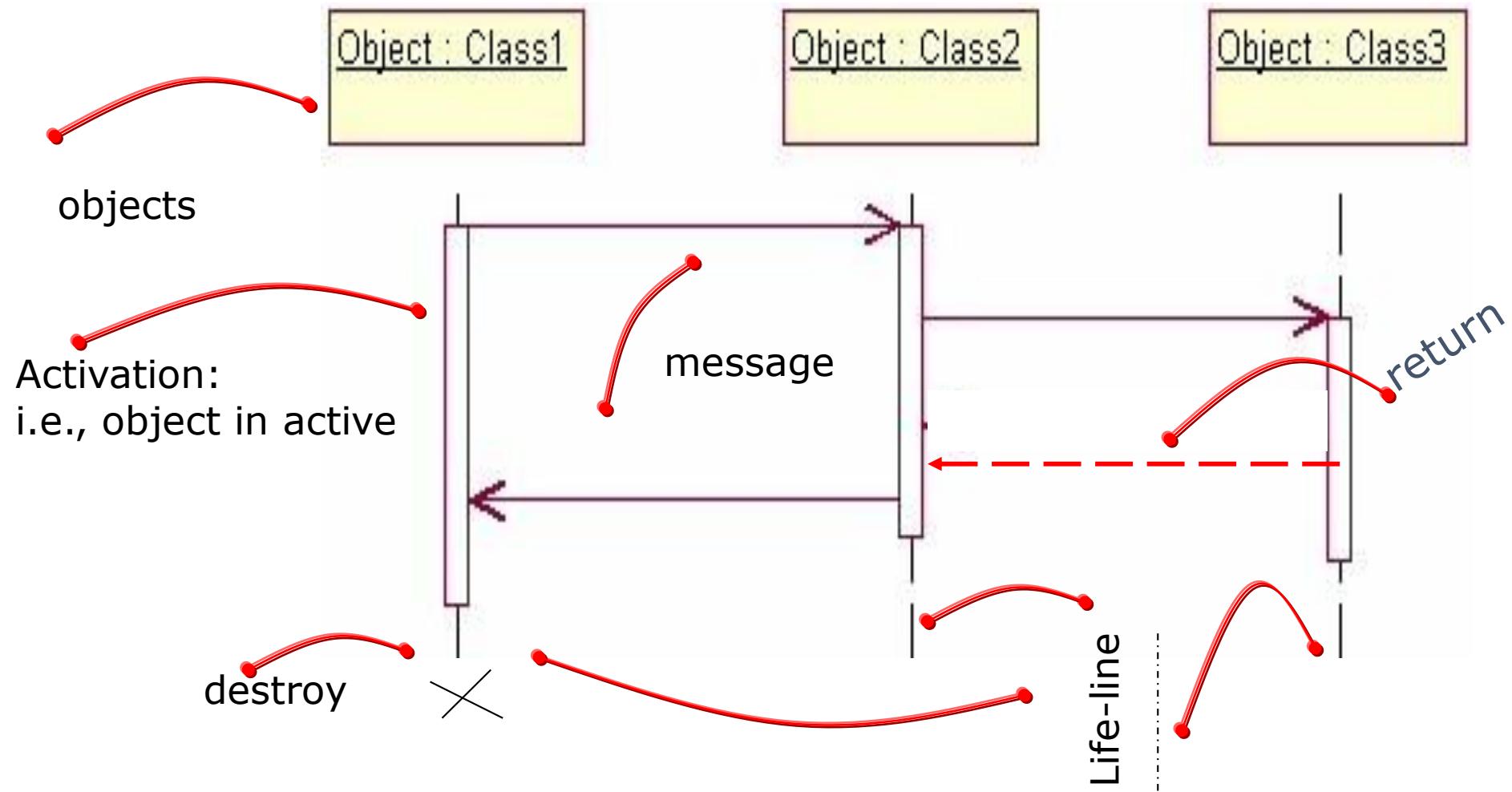
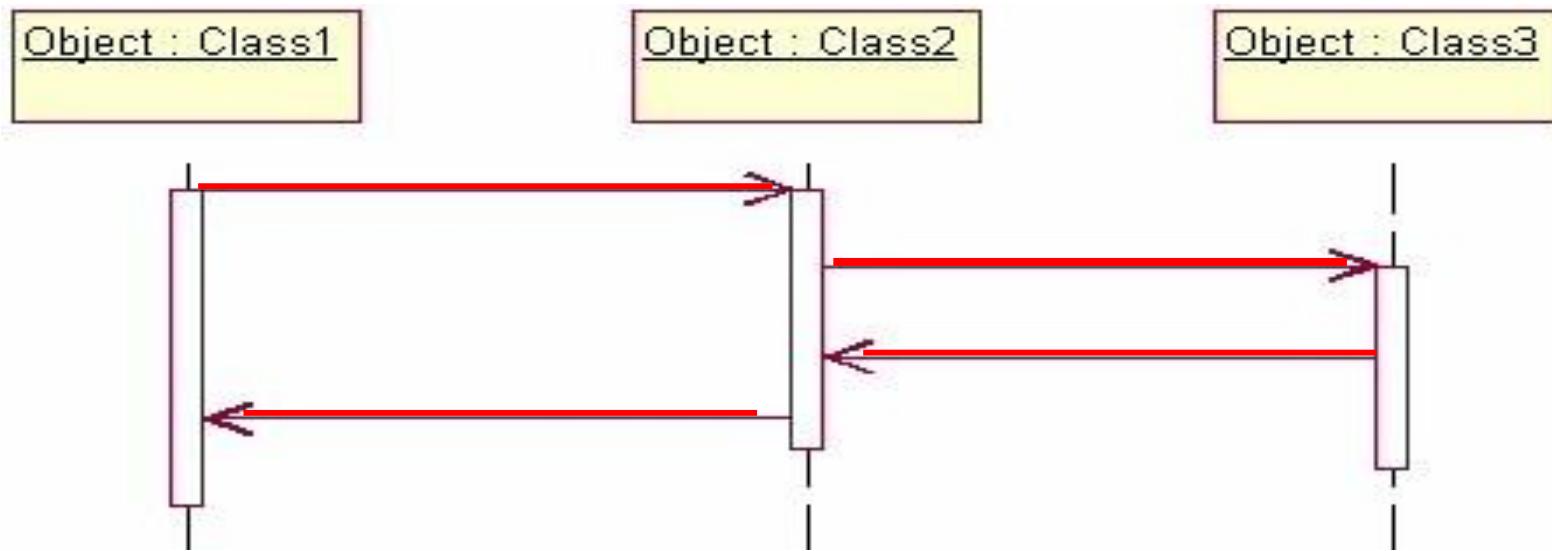


Diagrama de Secuencia



- El ejemplo muestra que un objeto de la clase 1 inicia el comportamiento enviando un mensaje a un objeto de la clase 2. Los mensajes pasan entre los diferentes objetos hasta que el objeto de la clase 1 recibe el mensaje final.

Ejemplo

- Máquina de autoservicio, tres objetos hacen el trabajo que nos preocupa
 - **the front**: la interfaz que la máquina de autoservicio presenta al cliente
 - **the money register**: parte de la máquina donde se recolecta el dinero
 - **the dispenser**: que entrega el producto seleccionado al cliente

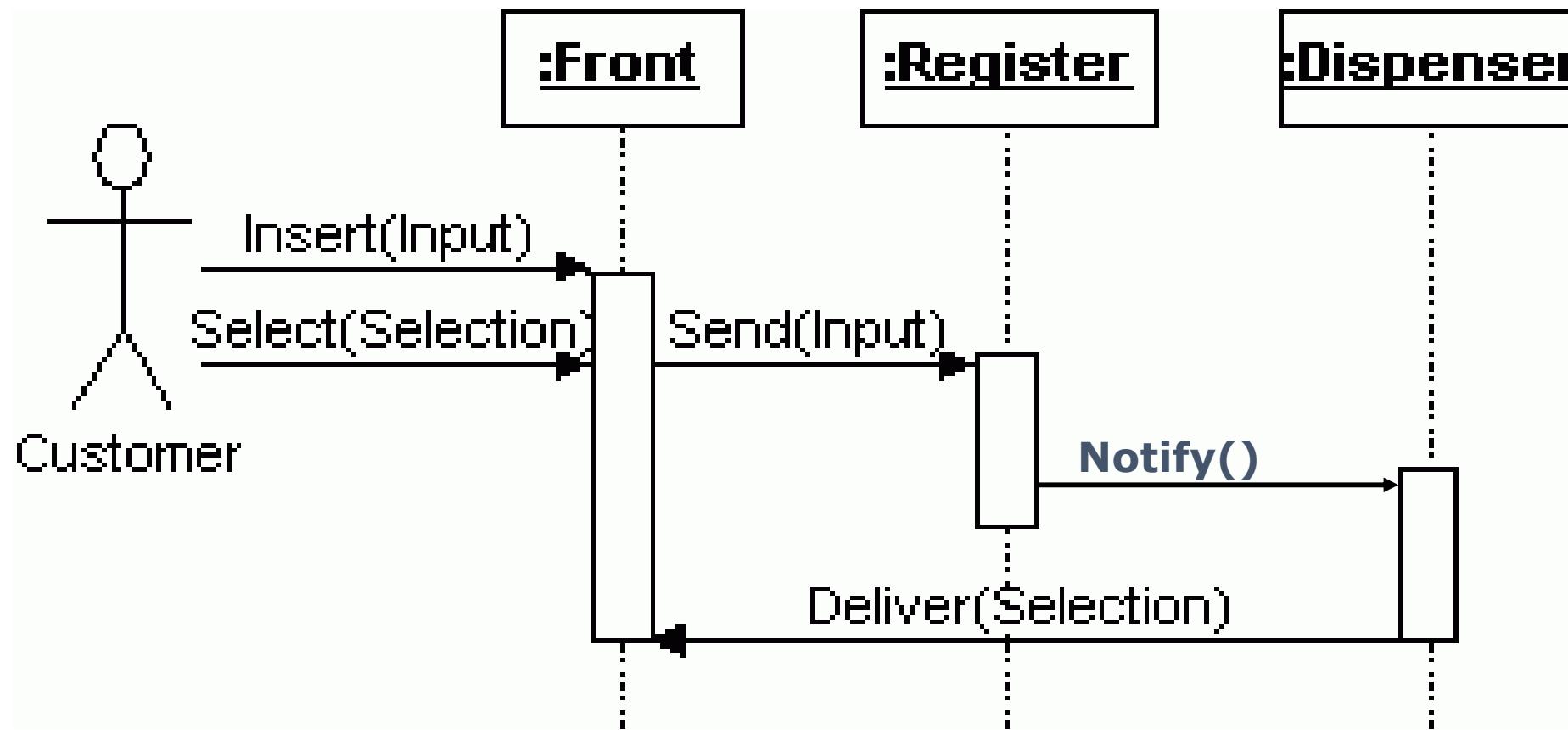


Ejemplo

- El diagrama de secuencia de instancias se puede dibujar usando estas secuencias:
 1. El cliente inserta dinero en la ranura de dinero
 2. El cliente hace una selección
 3. El dinero viaja a la caja registradora.
 4. El registro verifica si el producto seleccionado está en el dispensador
 5. El registro actualiza su reserva de efectivo
 6. La caja registradora tiene un dispensador que entrega el producto al frente de la máquina



Ejemplo



The "Buy a product" scenario.

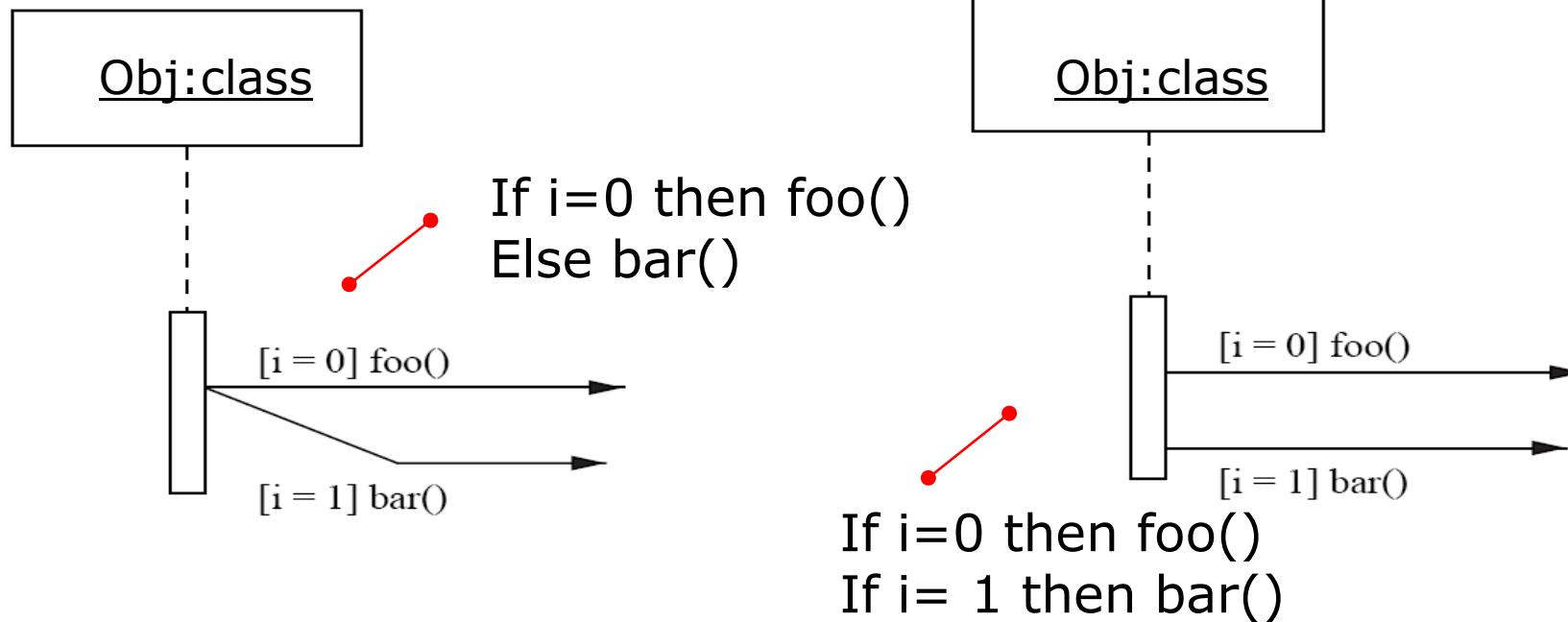
*Because this is the best-case scenario, it's an **instance sequence diagram***

... Pero

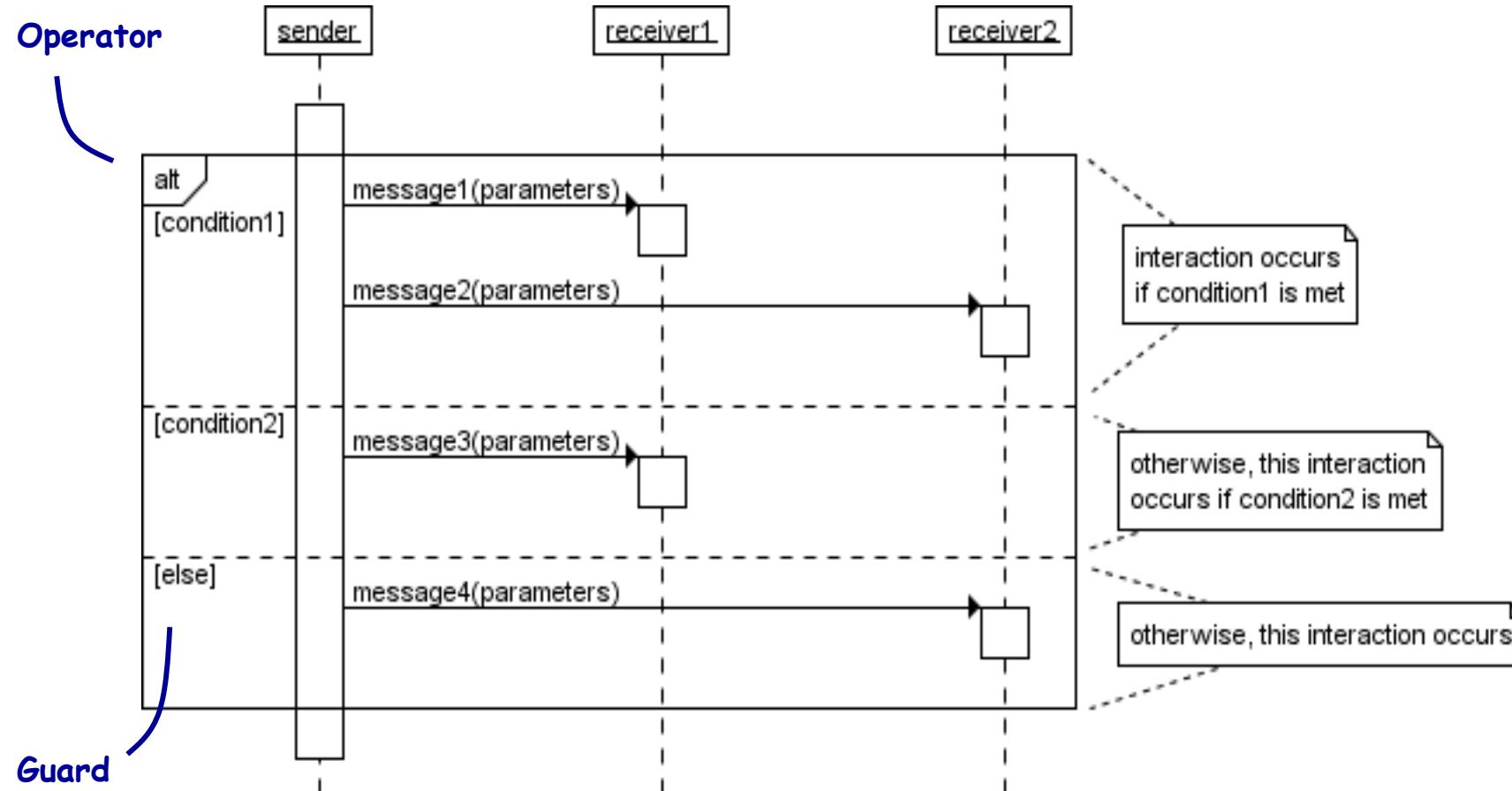
- Hemos visto una instancia de un diagrama de interacción:
una posible secuencia de mensajes
- Dado que un caso de uso puede incluir muchos escenarios
 - Es necesario mostrar un comportamiento condicional.
 - Es necesario mostrar posibles iteraciones
- Un diagrama de interacción genérico muestra todas las secuencias posibles de mensajes que pueden ocurrir

Mostrando comportamiento condicional

- Un mensaje puede estar **protegido** por una condición
- Los mensajes solo se envían si la **protección se evalúa como verdadera** en el momento en que el sistema alcanza ese punto en la interacción



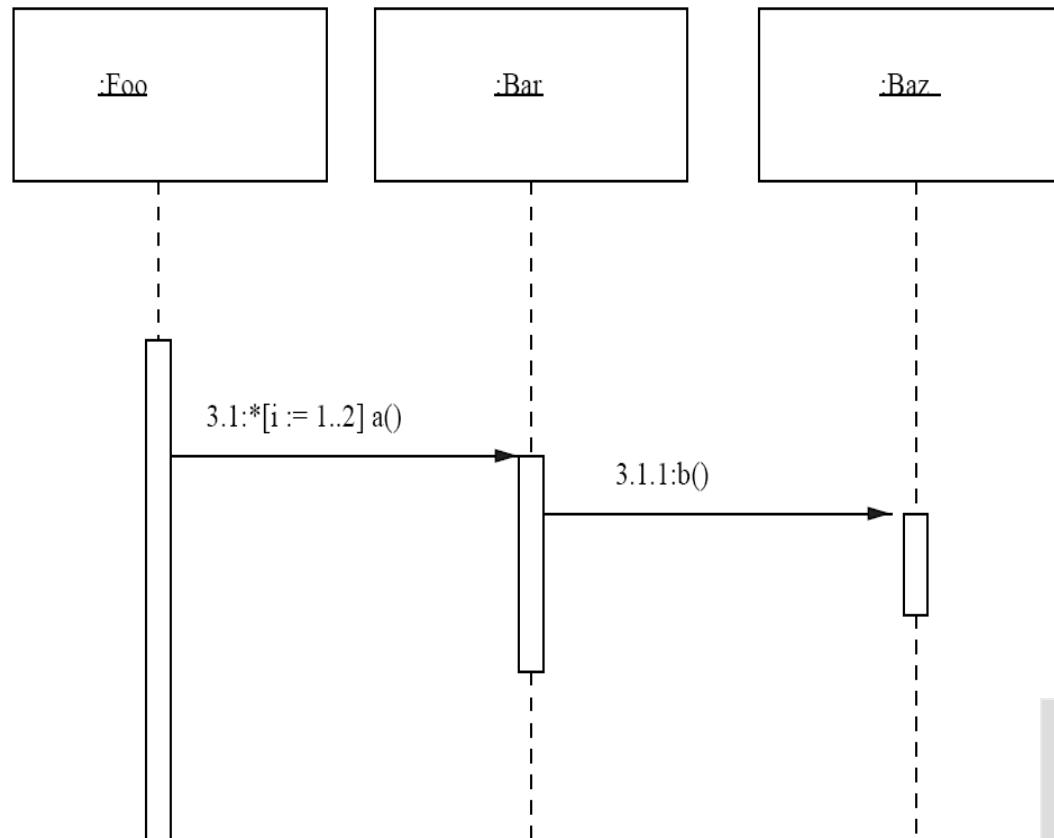
alt: Operadores en frames de interacciones – UML 2.0



Alternative multiple fragment: only the one whose condition is true will execute

Iteraciones (es decir, loop) – UML 1.0

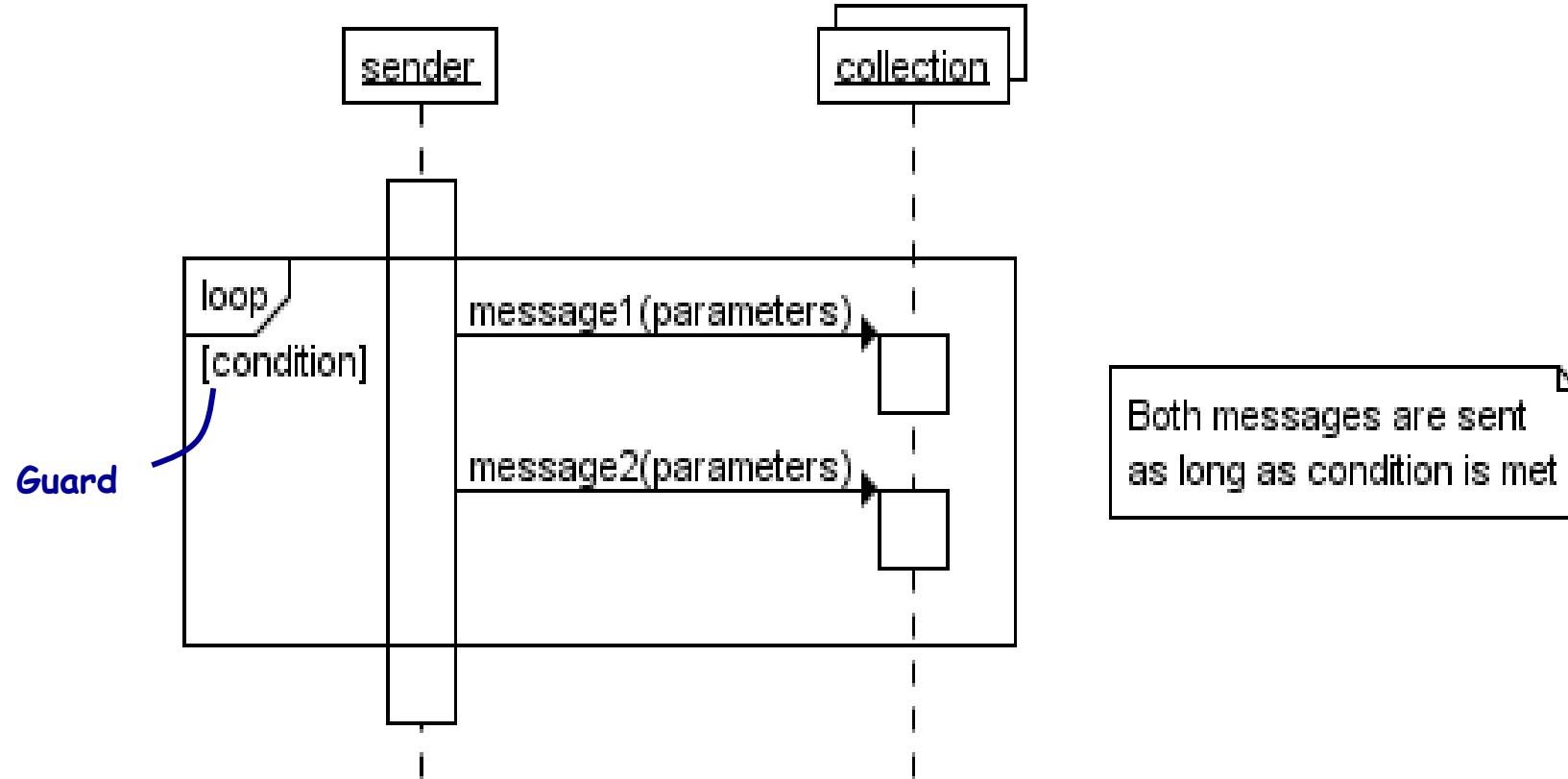
- * Indicates looping or iterations
- i:=1..2 means 2 iterations....



If you have seen it?
Earlier UML versions: UML
1.0

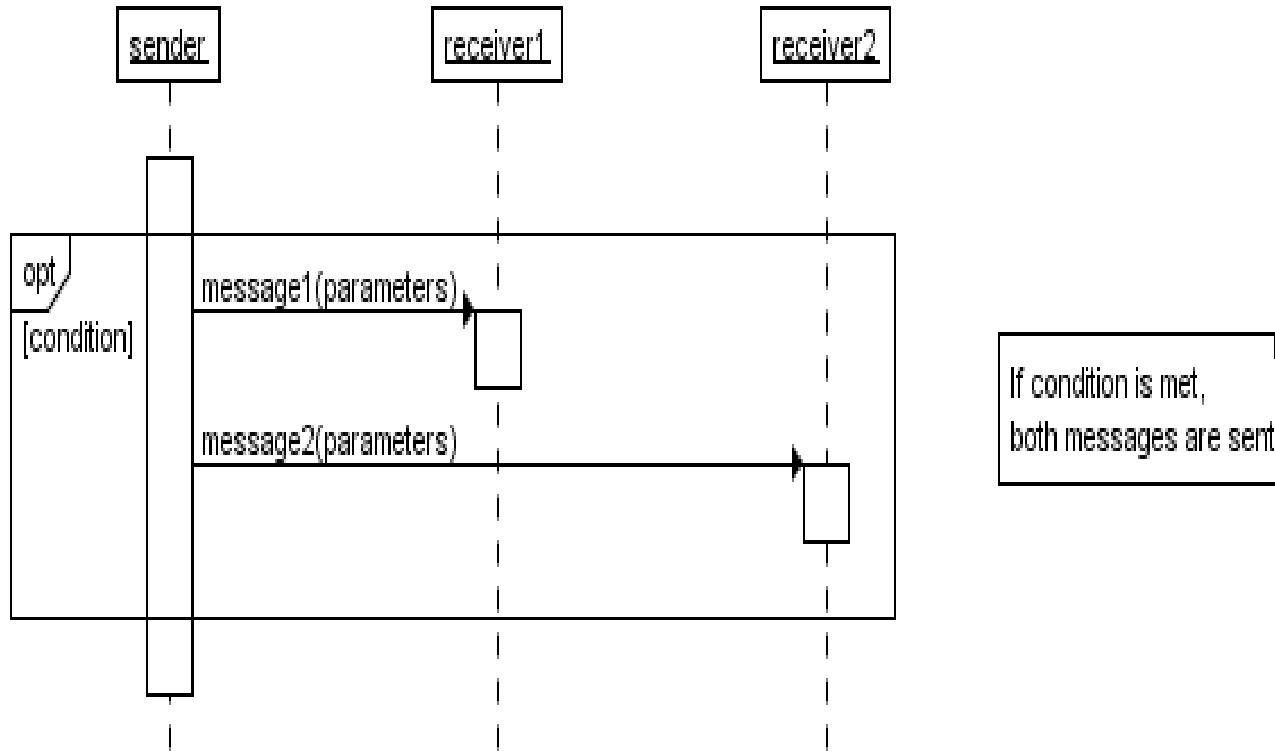
Result: ab
ab

Loop en UML 2.0



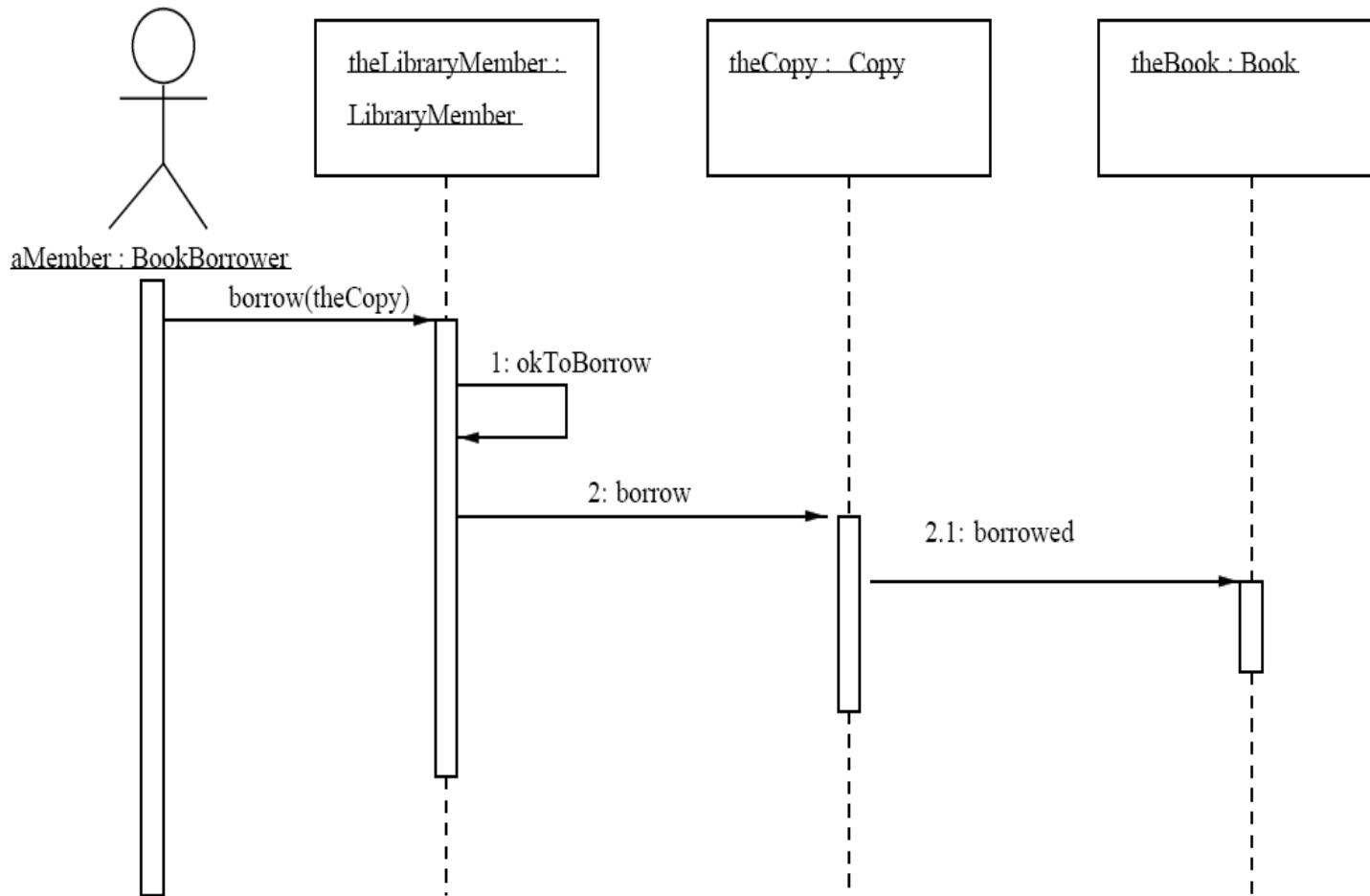
Loop: the fragment may execute multiple times, and the guard indicates basis for iterations

Opt en UML 2.0

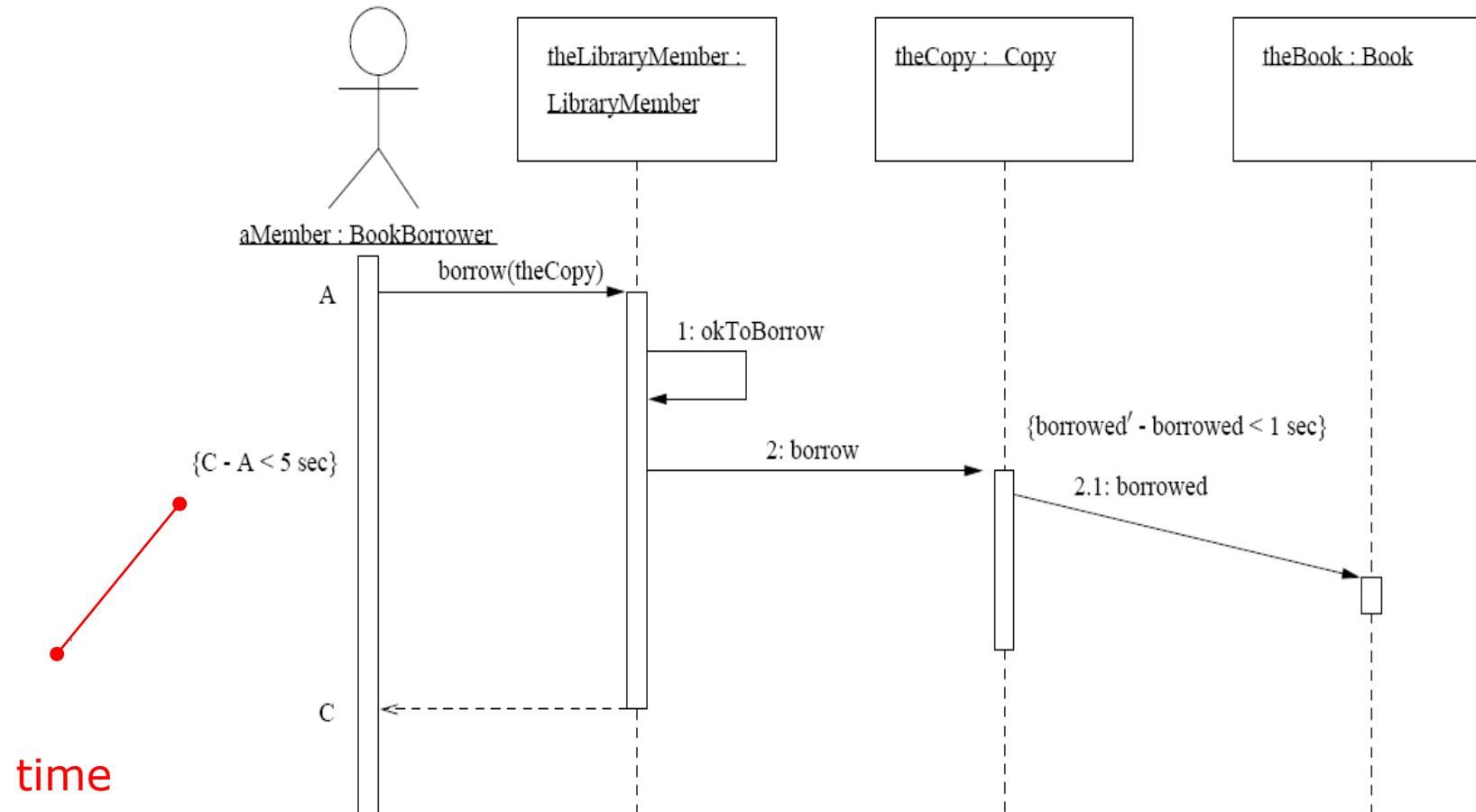


**Opt:Optional; the fragment executes only if the supplied condition is true.
This is equivalent to an alt with one trace**

Diagrama de Secuencia de la Biblioteca



Mostrando restricciones de tiempo en un diagrama



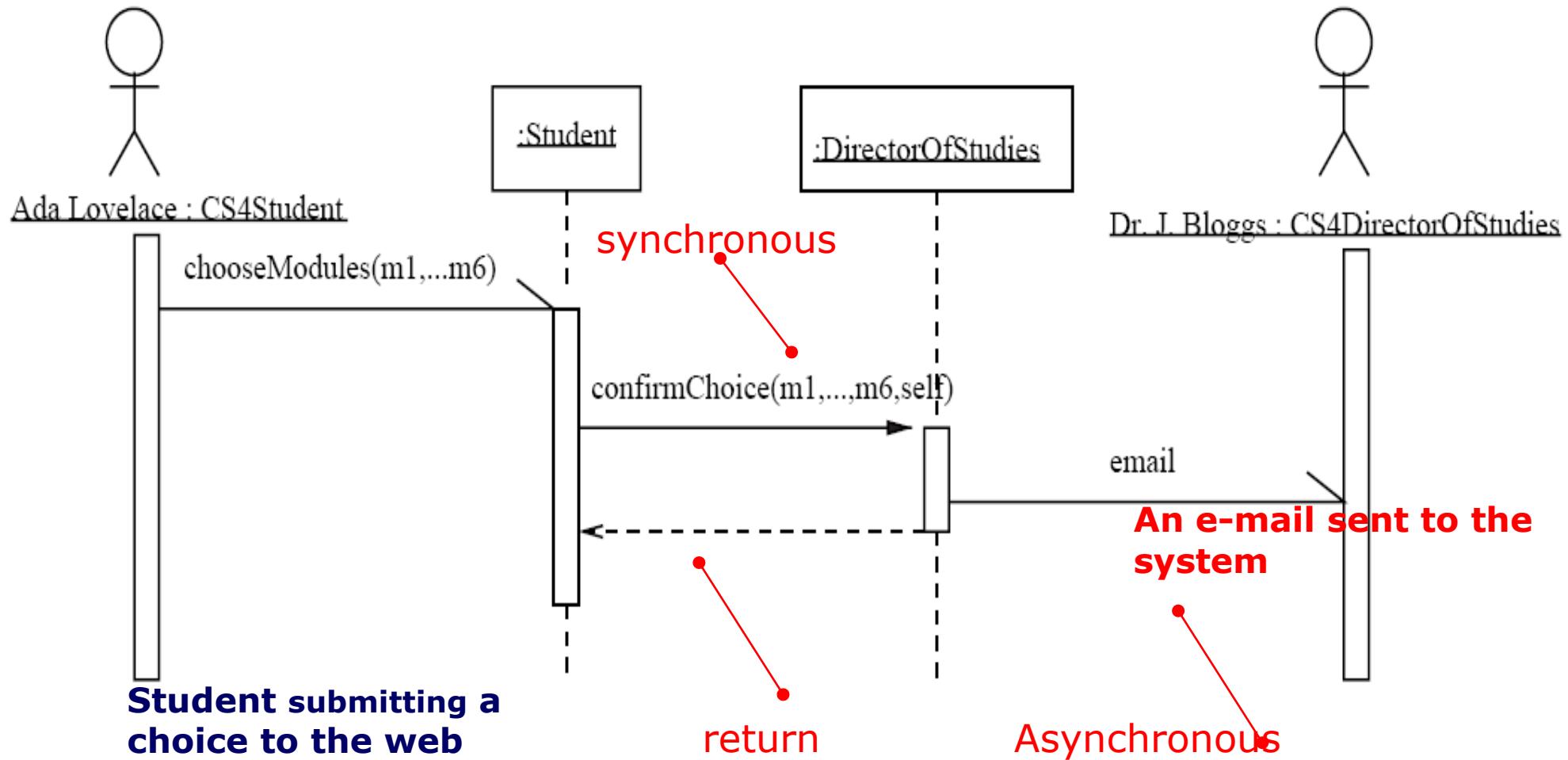
Tipos de Interacciones en Diagramas de Secuencia

Some UML
versions use
for both



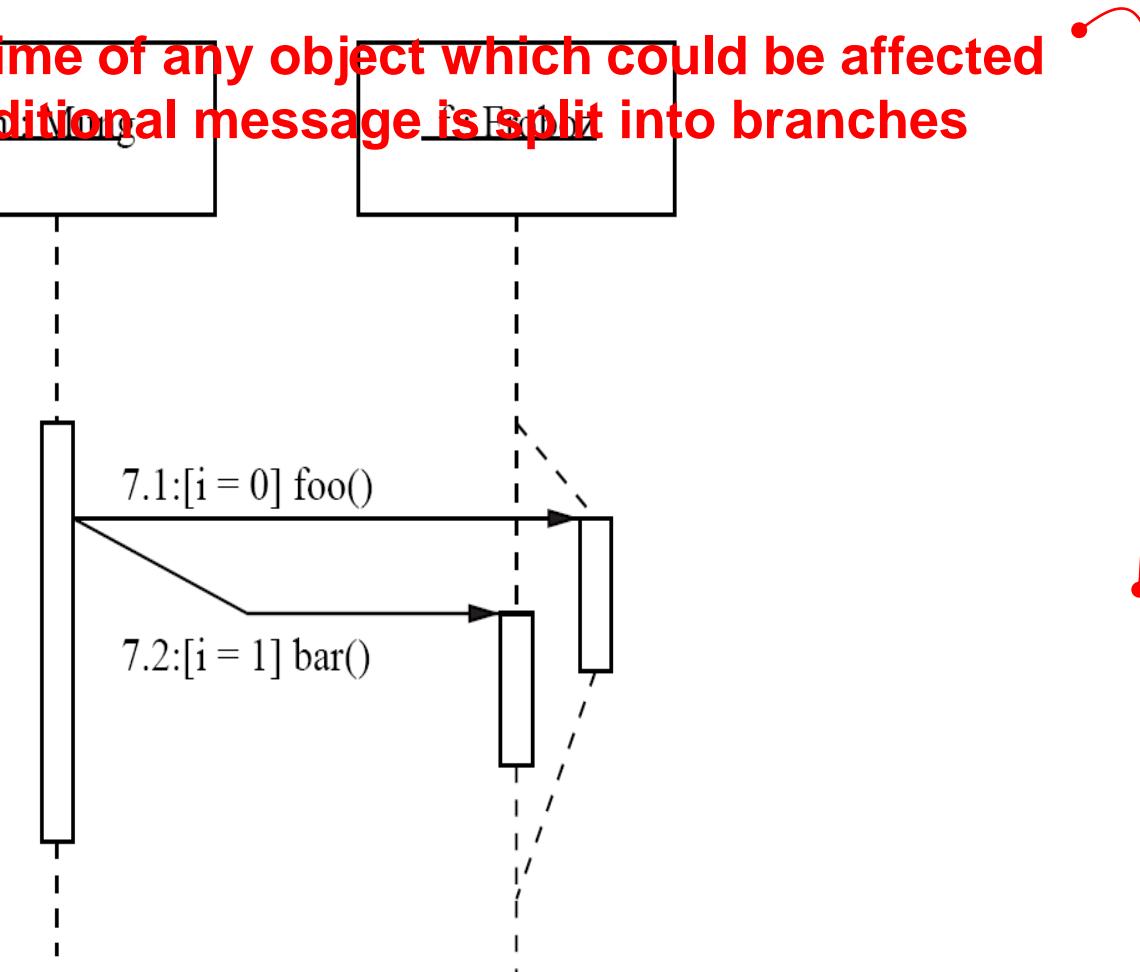
Interaction type	Symbol	Meaning
Synchronous or call	→	The ‘normal’ procedural situation. The sender loses control until the receiver finishes handling the message, then gets control back, which can optionally be shown as a return arrow.
Return	<--	Not a message, but a return from an earlier message. Unblocks a synchronous send.
Flat	→	The message doesn’t expect a reply; control passes from the sender to the receiver, so the next message (in this thread) will be sent by the receiver of this message.
Asynchronous	→	The message doesn’t expect a reply, but unlike the flat case, the sender stays active and may send further messages.

Ejemplo

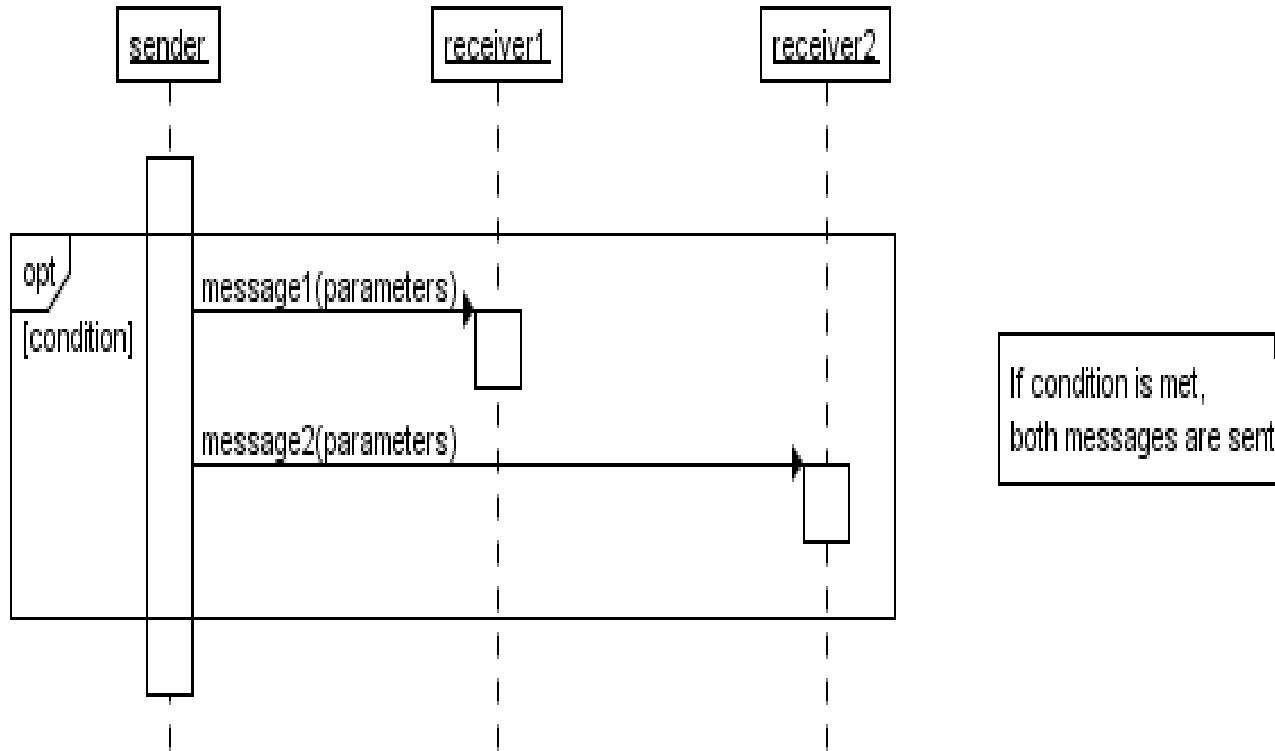


Other notions: Branching

The life time of any object which could be affected by a conditional message is split into branches



Opt en UML 2.0



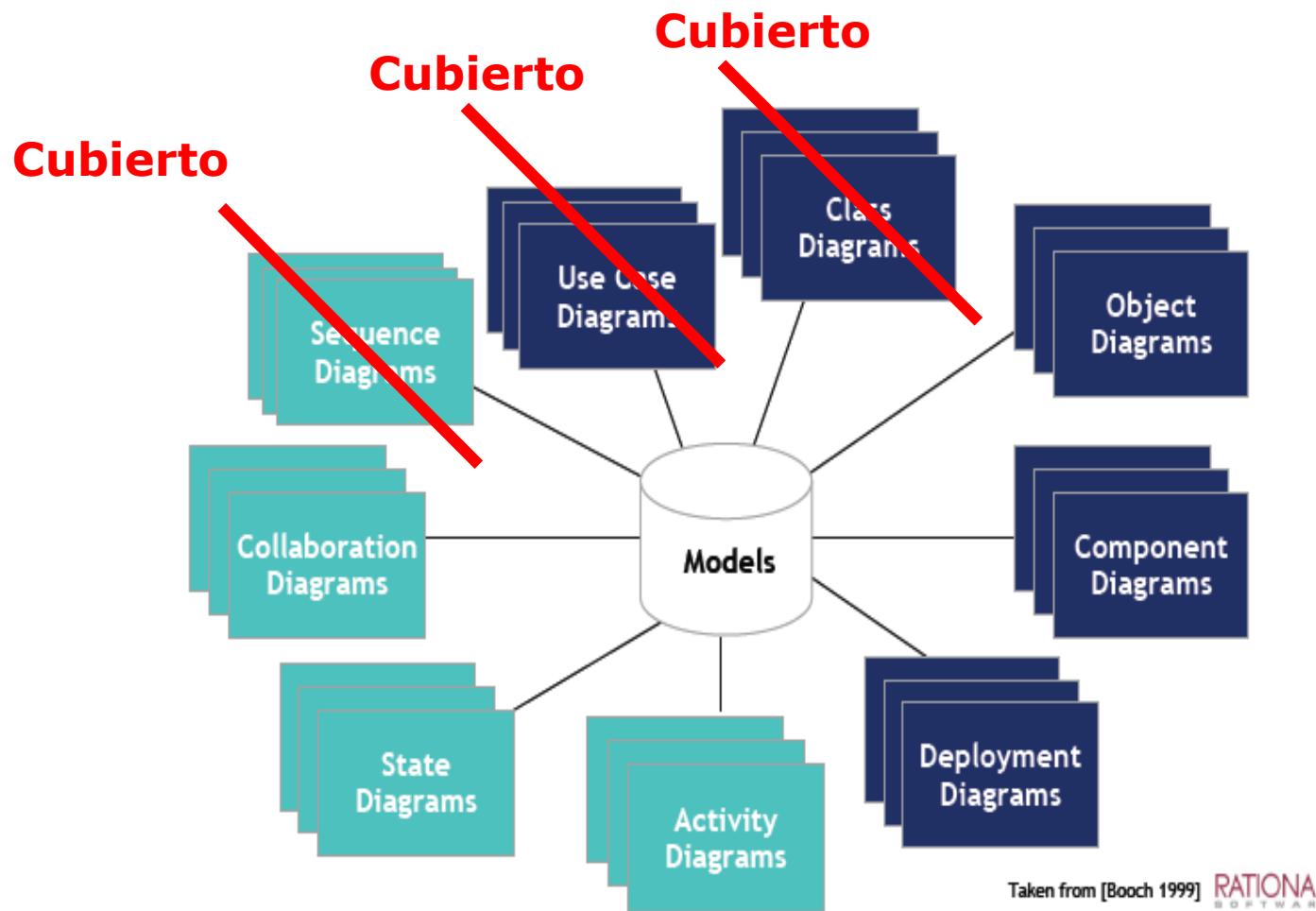
**Opt:Optional; the fragment executes only if the supplied condition is true.
This is equivalent to an alt with one trace**

Ejercicio

- Prepare de diagrama de casos de uso para un cajero automático
- Use un escenario de interés
- Dibuje un diagrama de secuencia correspondiente a los casos de uso.



Diagramas UML



Antes de finalizar

Puntos para recordar

- ¿Para qué sirve UML?
- ¿Cuáles son los principales diagramas?
- ¿Cuál es la relación entre modelo y diagrama?
- Recuerde los elementos de cada diagrama UML
- Memorice ejemplos típicos de cada diagrama UML
- ¿Cuáles son los enfoques para identificar clases?

Lectura adicional

- Rumbaugh, Jacobson and Booch, “The Unified Modeling Language Reference Manual”
 - Chapters 1, 2,3, 4, 5 y 8
- Perdita Stevens with Rob Pooley, “Using UML”
 - Chapters 1, 2,3, 5, 6, 7, 8 y 9
- Pressman and Maxin , “Software Engineering”
 - Appendix 1: An Introduction to UML
 - Appendix 2: Object-Oriented Concepts

Próxima sesión

- Proceso de implementación a partir de un diseño.