

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

Semana 4

# Agenda

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

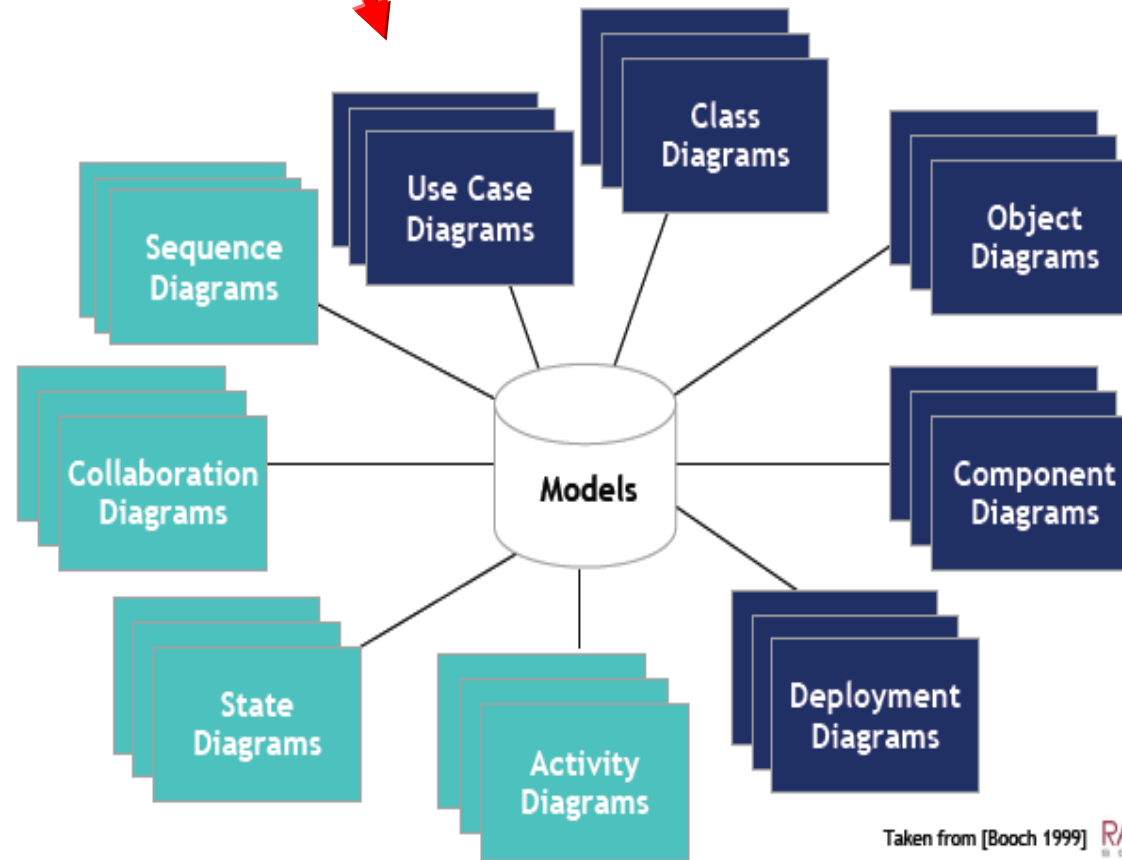
# Lenguaje unificado de modelado

# 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.

# Diagramas UML

**Estamos aquí!**



Taken from [Booch 1999] **RATIONAL** SOFTWARE

# Ejercicio – Sistema de registro académico

- Realice el Diagrama de Casos de Uso con al menos 2 refinamientos de cada tipo
  - includes, extends.
- Describa 2 escenarios de uso del sistema.
- Describa el caso de uso “Registrarse en un paralelo” realizado por un estudiante.

## **Registrarse en un paralelo**

Actores (Principales y secundarios)

1.

Precondiciones

1.

Flujo eventos (incluir alternativos)

1.

2.

Post condiciones

1.

2.

# Sistema de registro académico

El sistema AcademicChoice permite a los estudiantes gestionar la inscripción de materias, paralelos y horarios en el semestre actual. Cada materia puede ofrecer múltiples paralelos con diferentes horarios y profesores, y un estudiante puede inscribirse en un paralelo específico, sujeto a la disponibilidad de cupos. Un paralelo puede estar en los siguientes estados: abierto para inscripción, en lista de espera o cerrado por falta de cupos.

Los estudiantes pueden buscar materias según criterios como nombre, código de la materia, horario y profesor, para luego seleccionar el paralelo de su preferencia. El sistema debe garantizar que no haya conflictos de horarios en la inscripción del estudiante y que se respeten los pre-requisitos necesarios para cada materia. Una vez confirmada la inscripción, el sistema actualizará automáticamente el estado del cupo en el paralelo y enviará una notificación al estudiante sobre el estado de su inscripción.

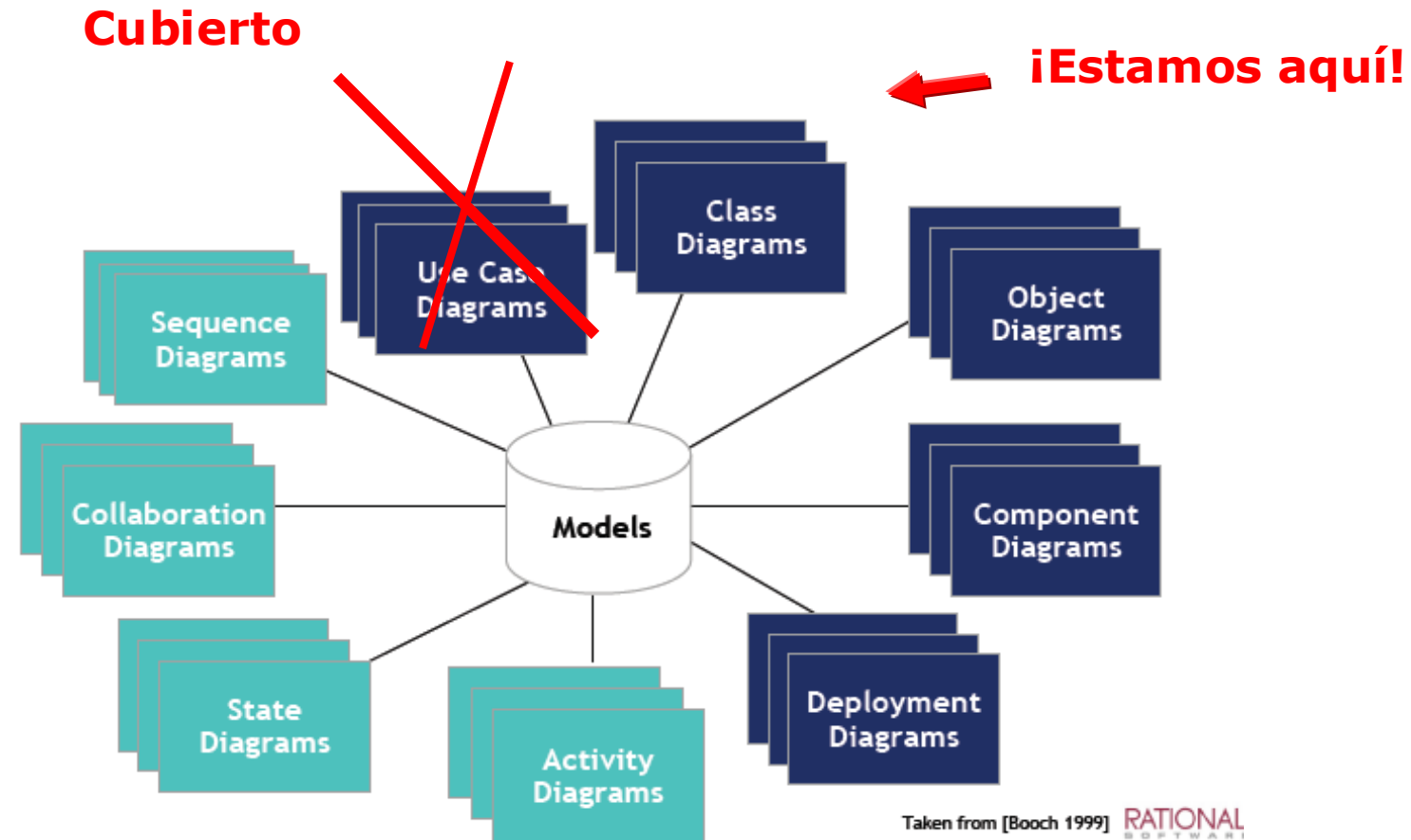
Los administradores del sistema pueden definir políticas de inscripción, tales como fechas límite para cambios y periodos de inscripción especiales para diferentes grupos de estudiantes (por ejemplo, estudiantes de último año). También pueden gestionar la apertura de nuevos paralelos o la reasignación de cupos en casos excepcionales.

El sistema ofrece notificaciones a los estudiantes sobre actualizaciones en sus inscripciones, cambios en horarios o profesores, y cualquier anuncio académico relevante, a través de canales configurables como correo electrónico, SMS o plataformas internas. Además, en caso de problemas con la inscripción o conflictos de horarios, los incidentes reportados son resueltos inicialmente por el personal de soporte académico y, si es necesario, se escalan al departamento de registro para una solución definitiva.

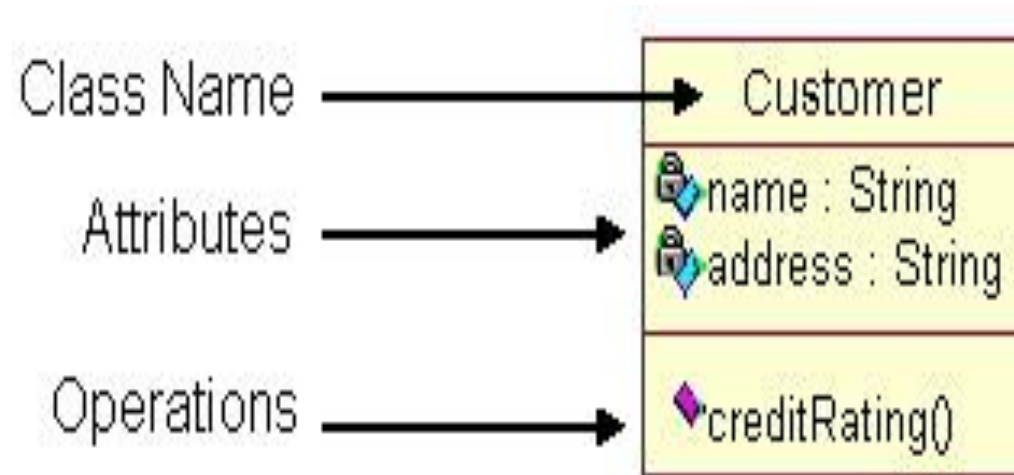
# Diagrama de clases



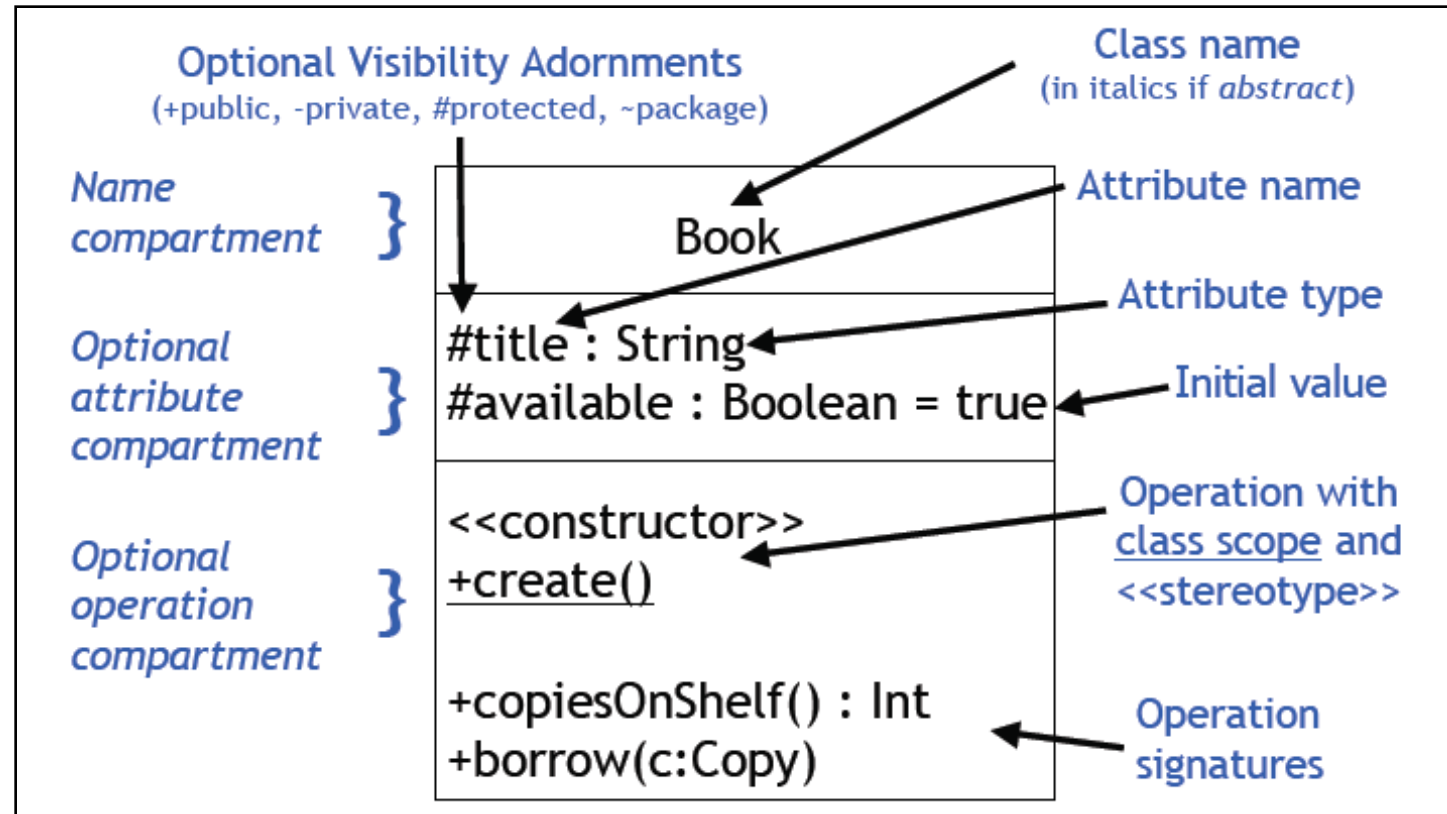
# Diagramas UML



# 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

- 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

# Ejemplo

- **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.

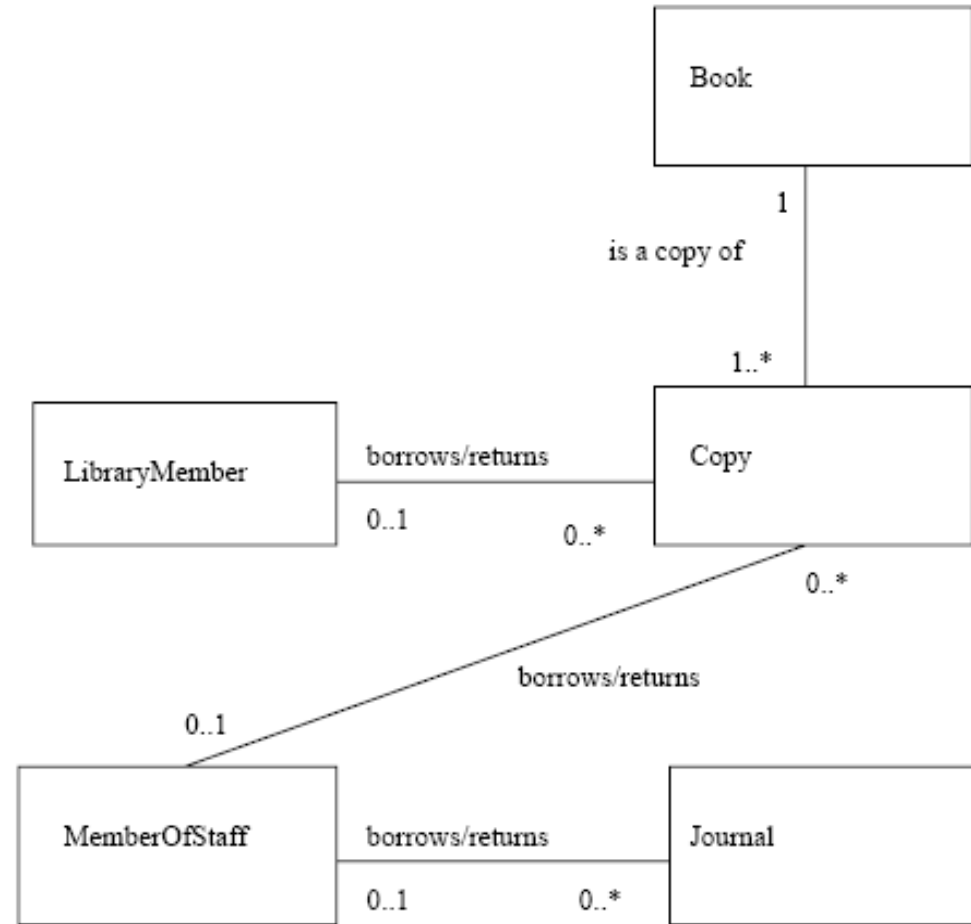
# Análisis sustantivo/verbo

**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.

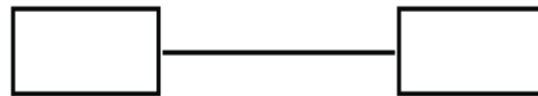
**Borrowing** The system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.



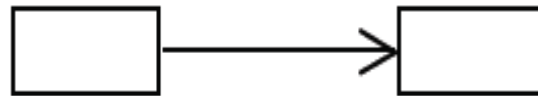
# First-Cut de Diagrama de Clases



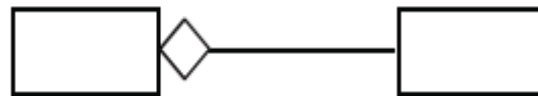
# Notación de Relaciones en UML



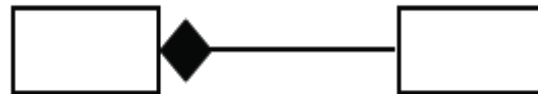
bidirectional / binary



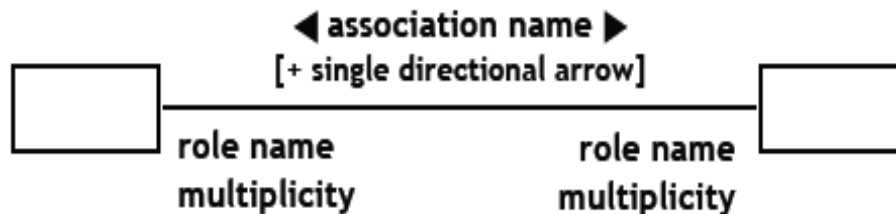
unidirectional



aggregation



composition

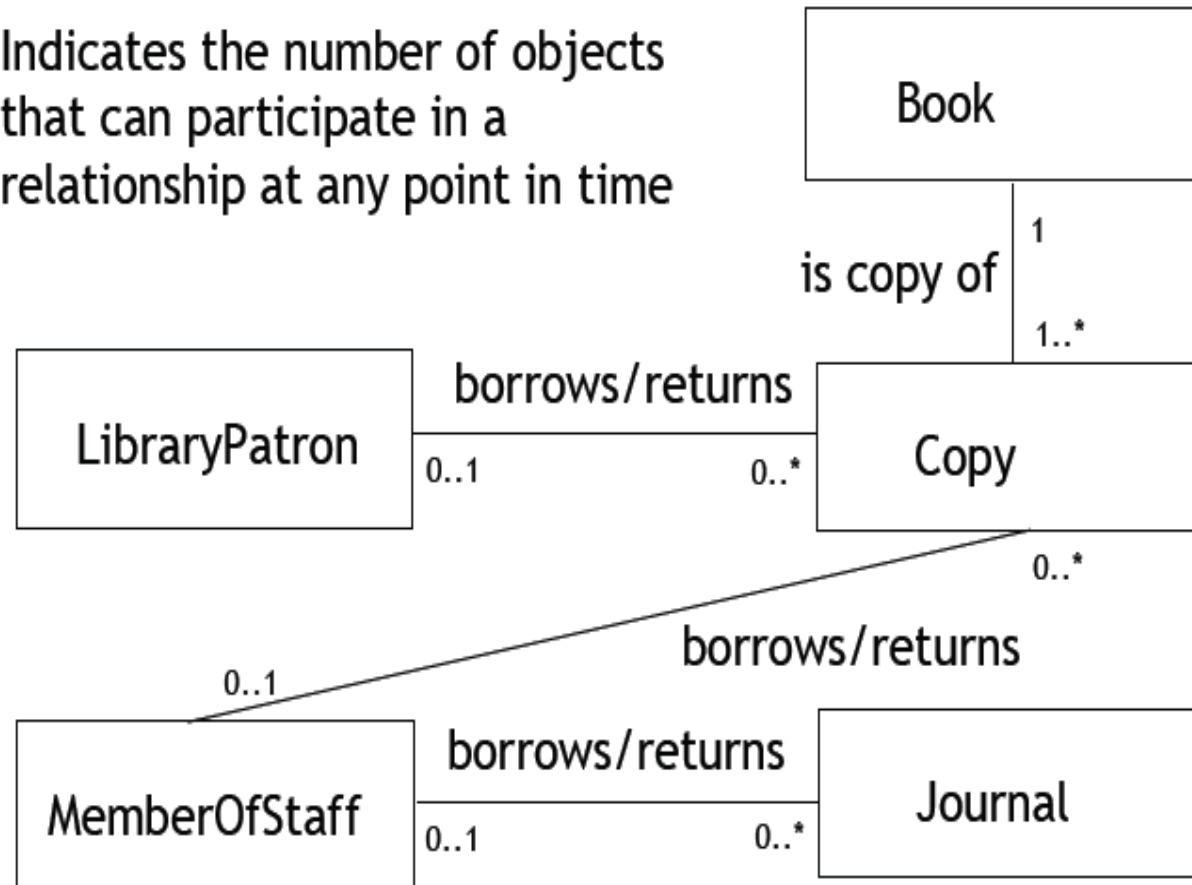


supplementary  
characteristics

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

# Asociaciones Parte/Todo

- **Aggregation:** Weak Ownership

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

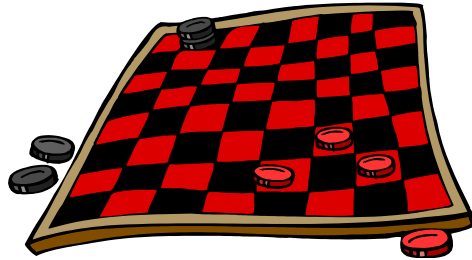


A module is part of a course

In fact,

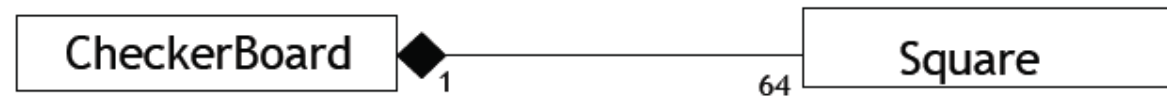
5 or more modules are part of one or more courses

# 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!

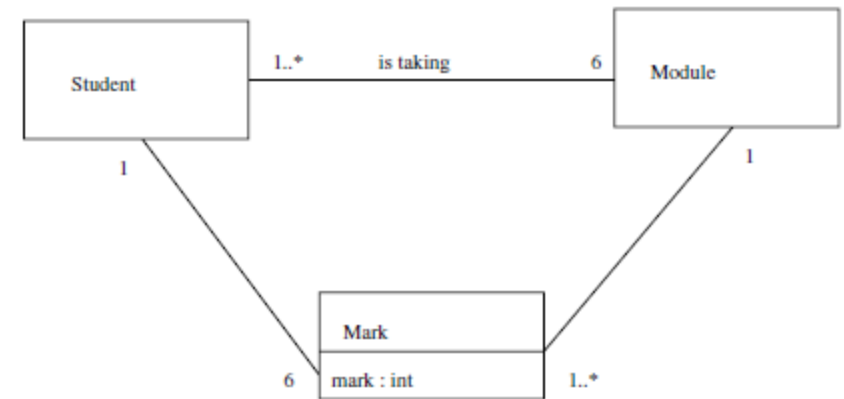
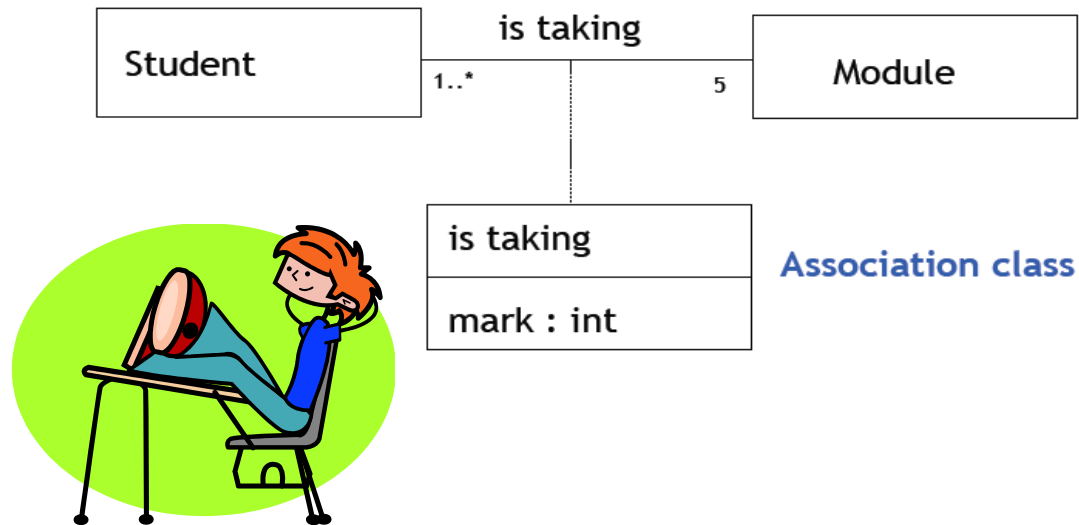
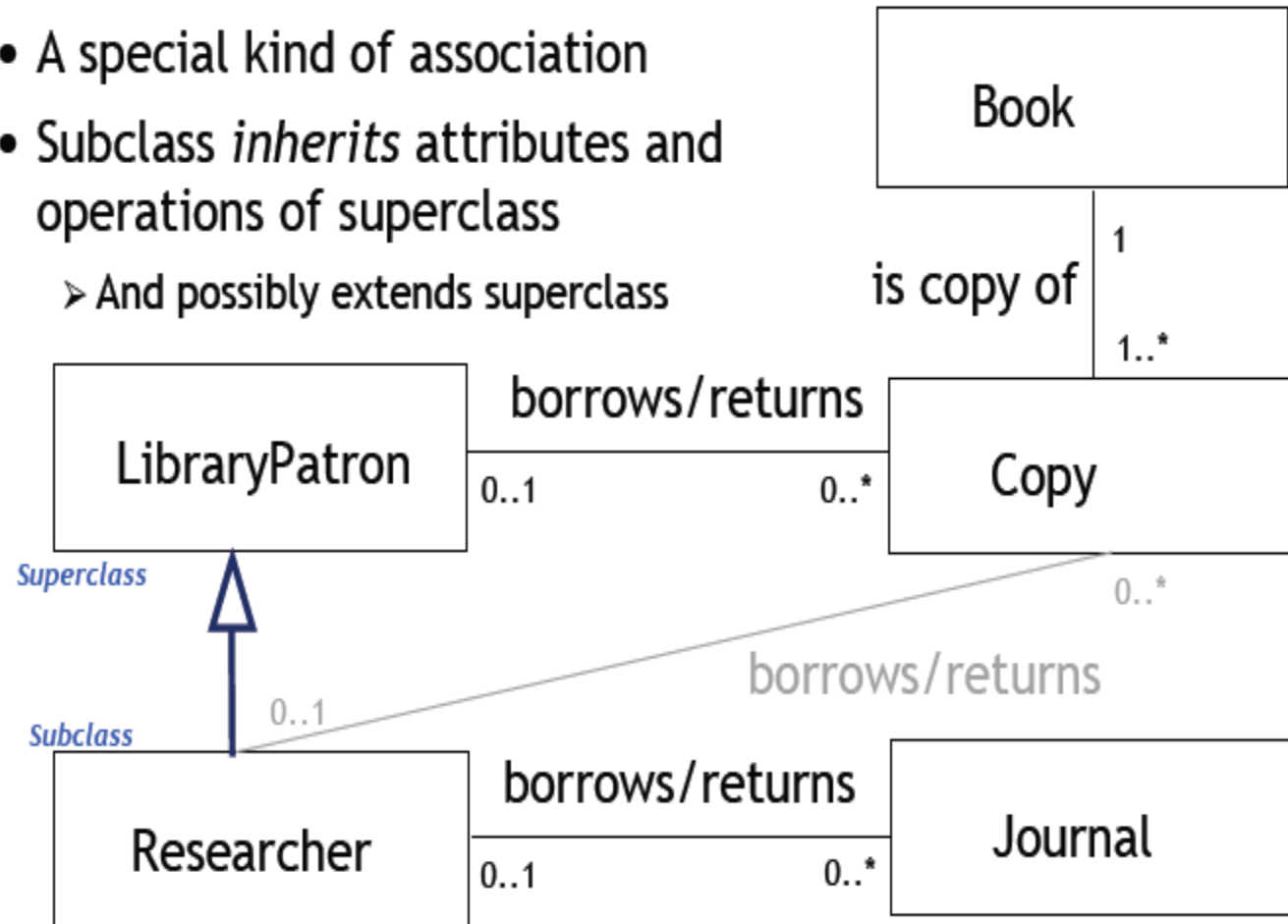


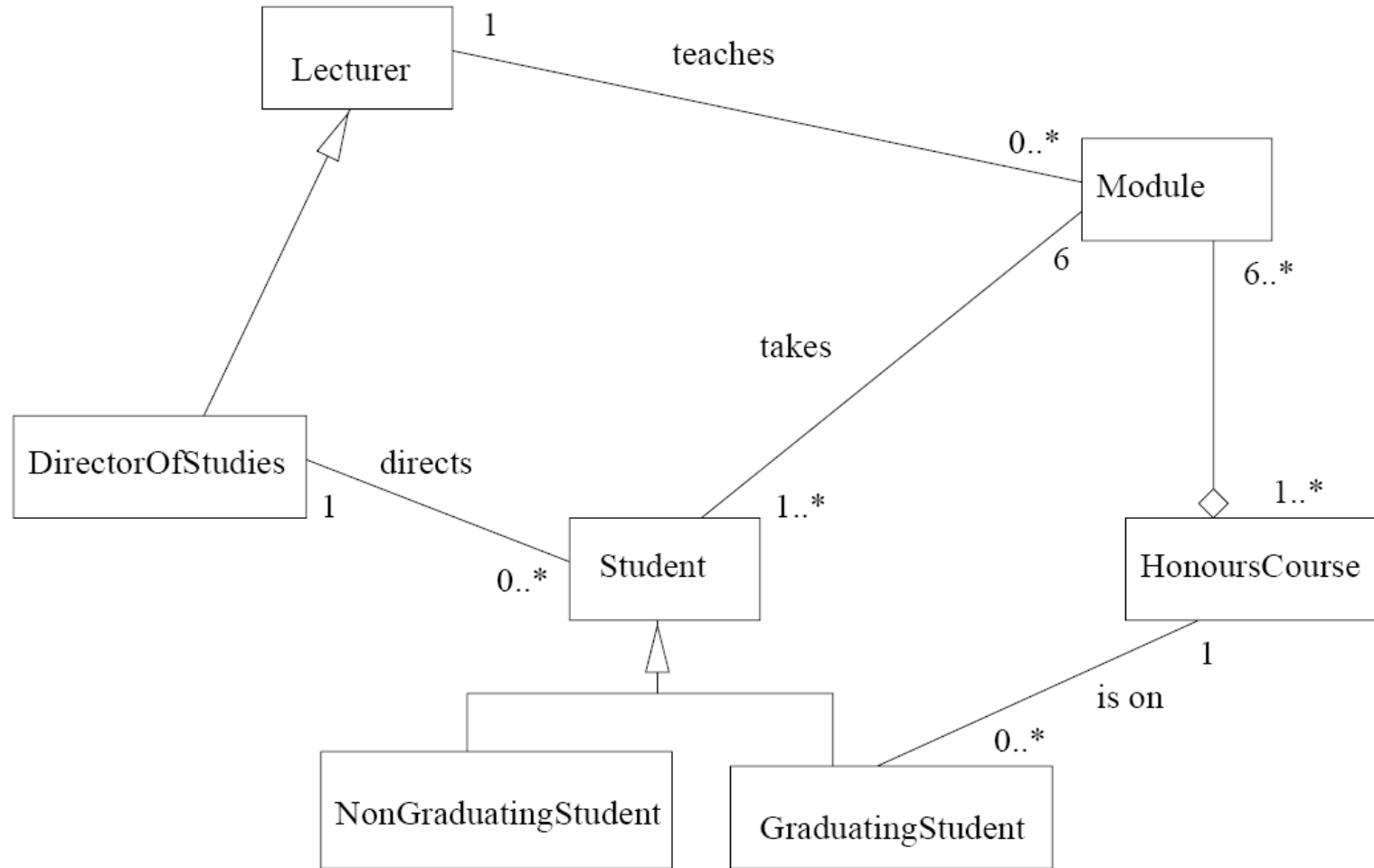
Figure 6.13 Avoiding an association class.

# Generalización y Herencia

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

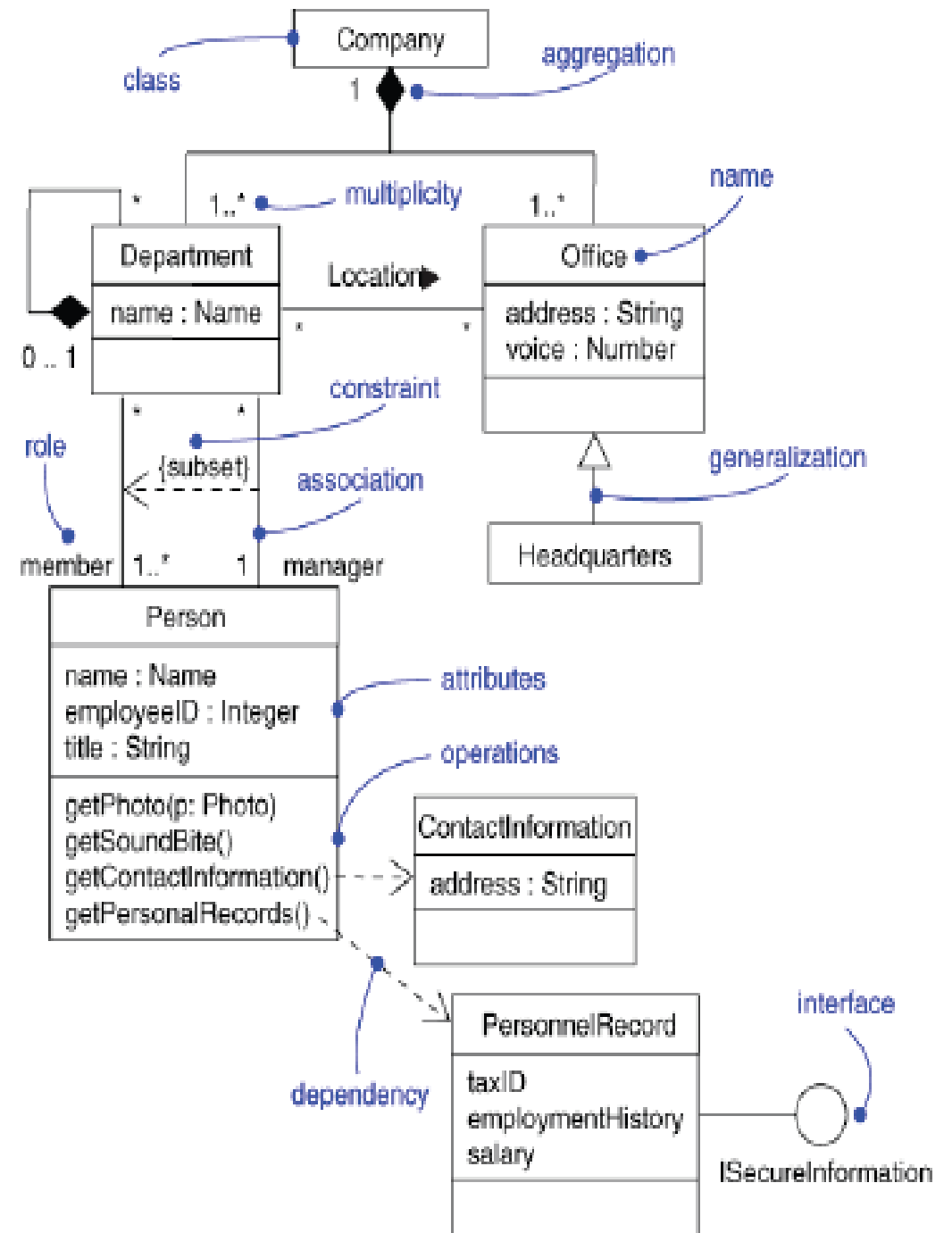


# Ejemplo: Modelo de Clases

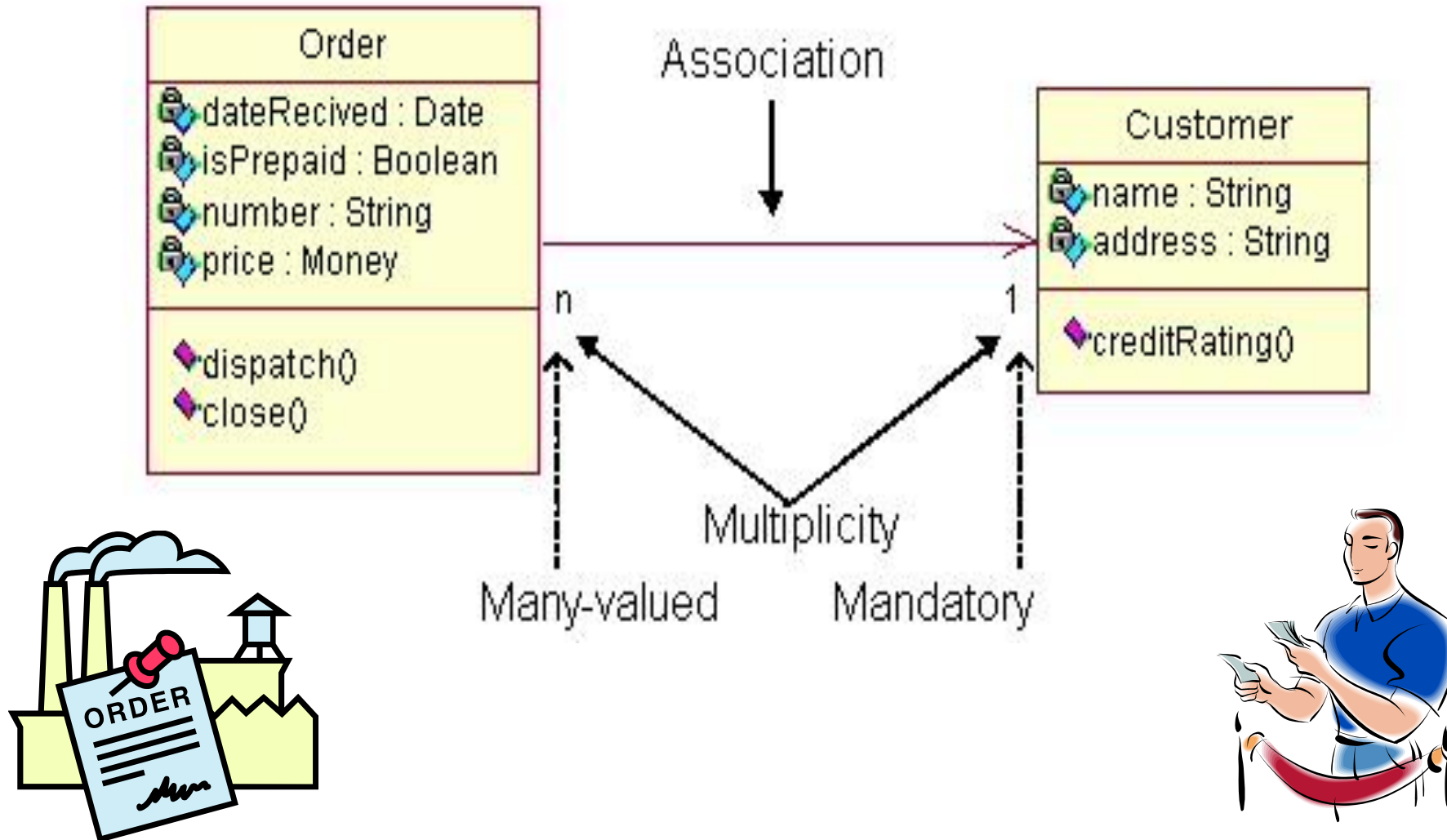




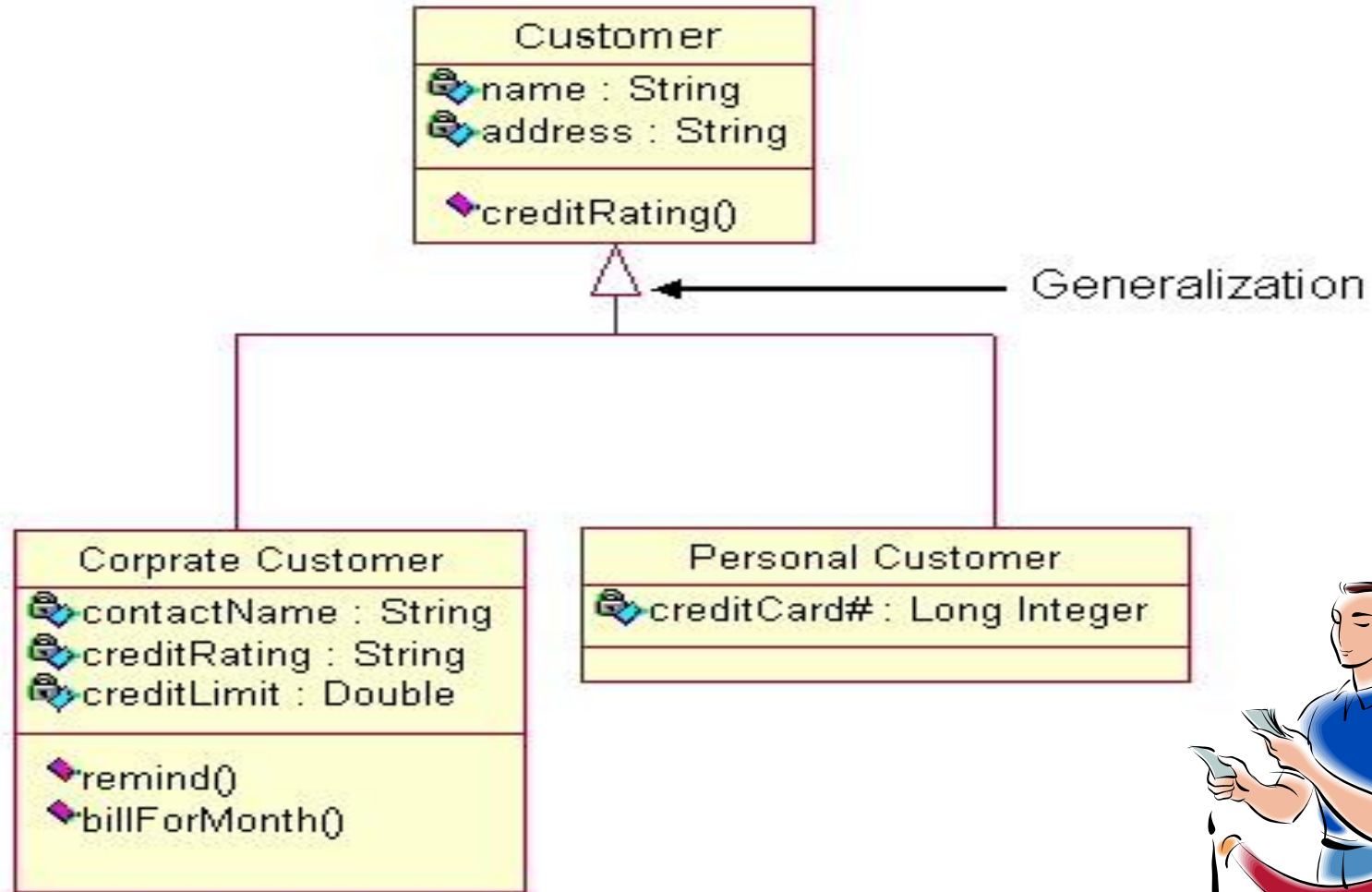
# Ejemplo: Diagrama de



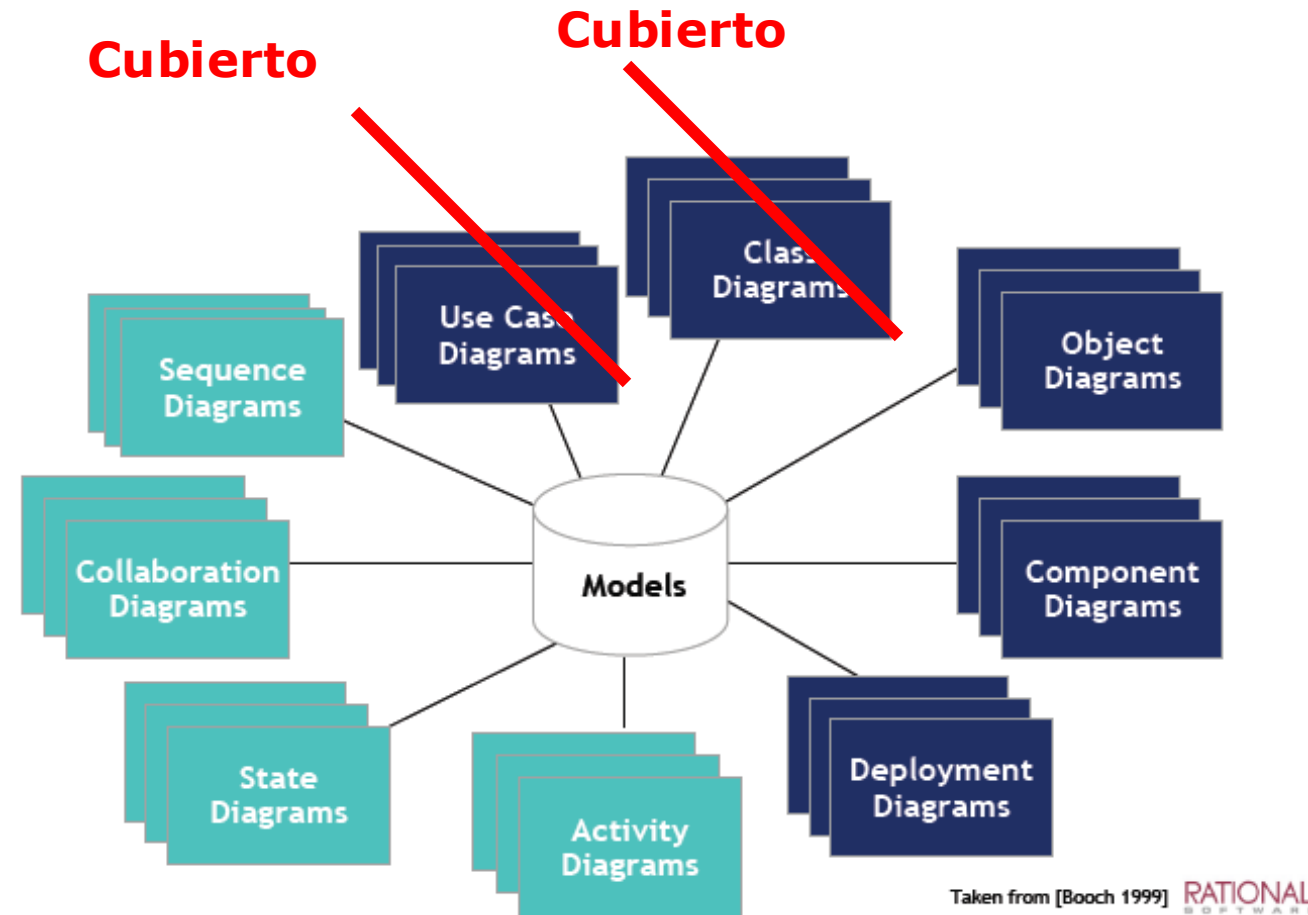
# More Examples



# Más ejemplos



# Diagramas UML

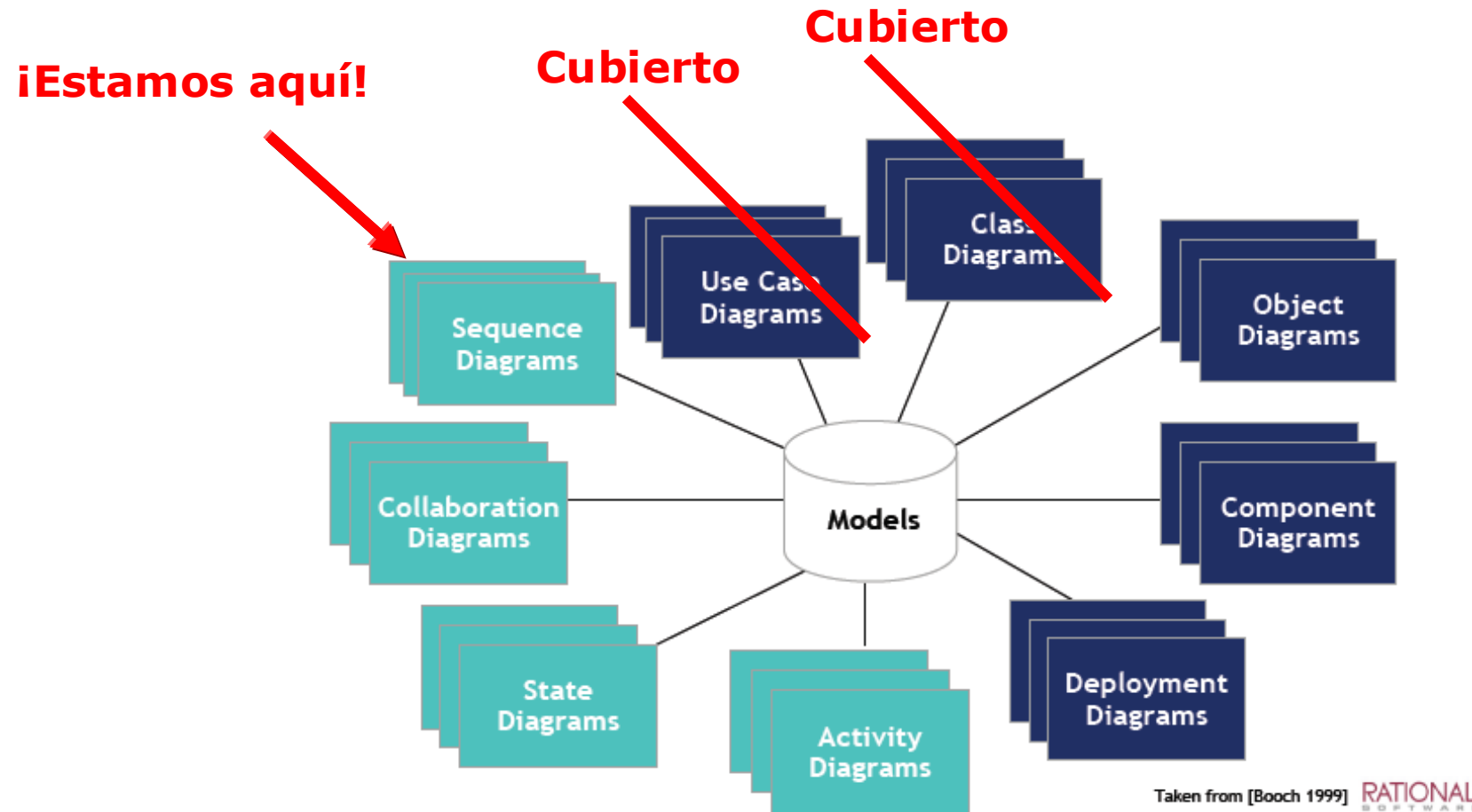


# En nuevos grupos de 3 personas

- Realice un diagrama de clases de First-cut
- Indique:
  - Nombres de clases
  - Relaciones entre clases
  - Nombres de las relaciones
  - Multiplicidad
  - Métodos o atributos importantes

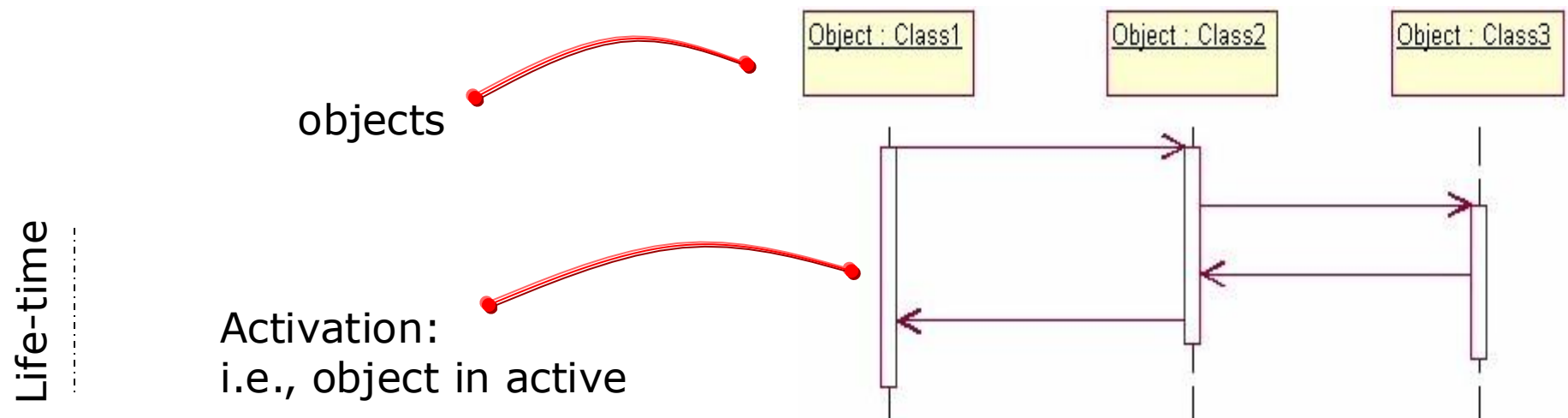
# Diagrama de secuencia

# Diagramas UML



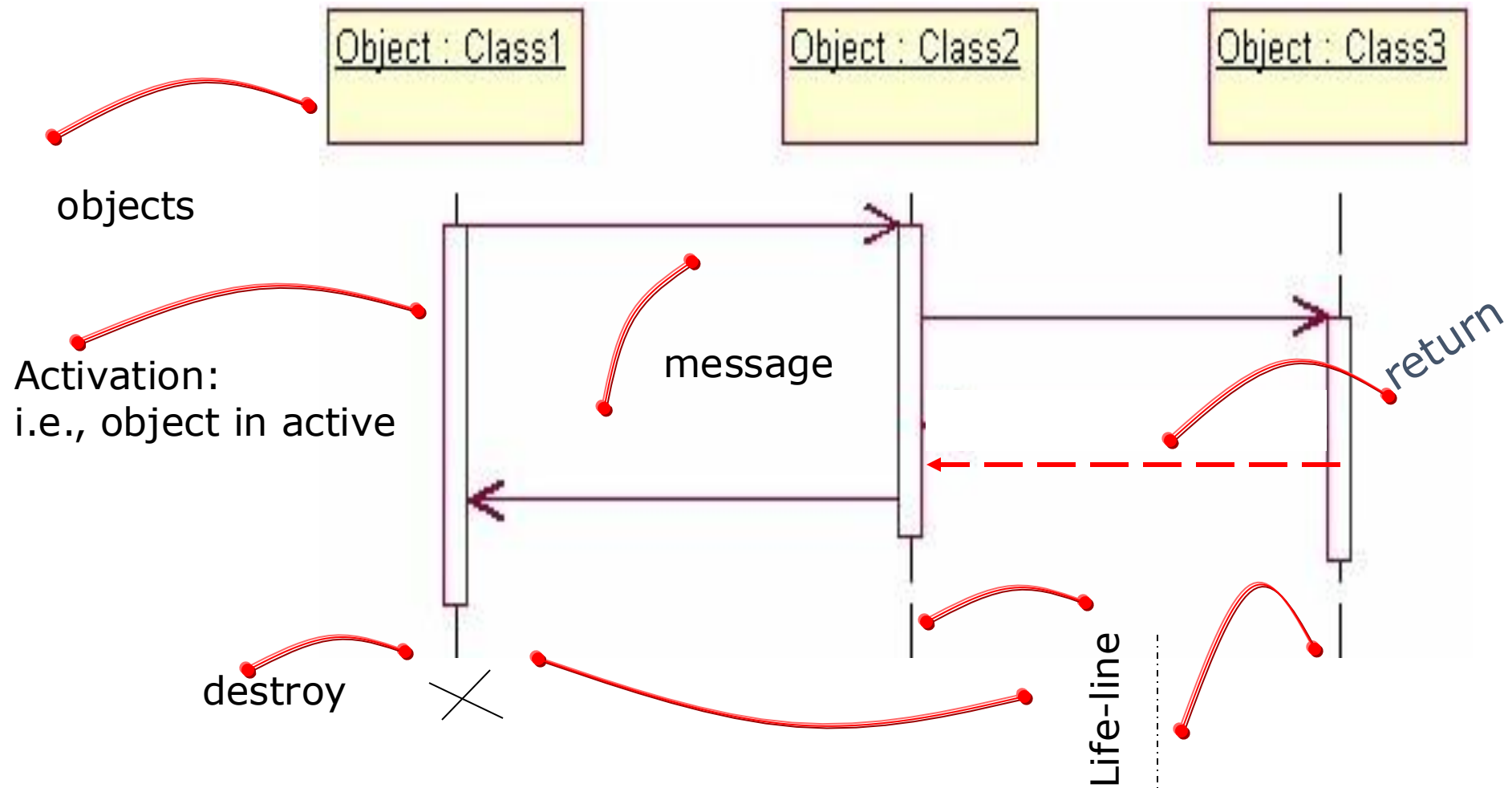
# Diagrama de Secuencia

- 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**)

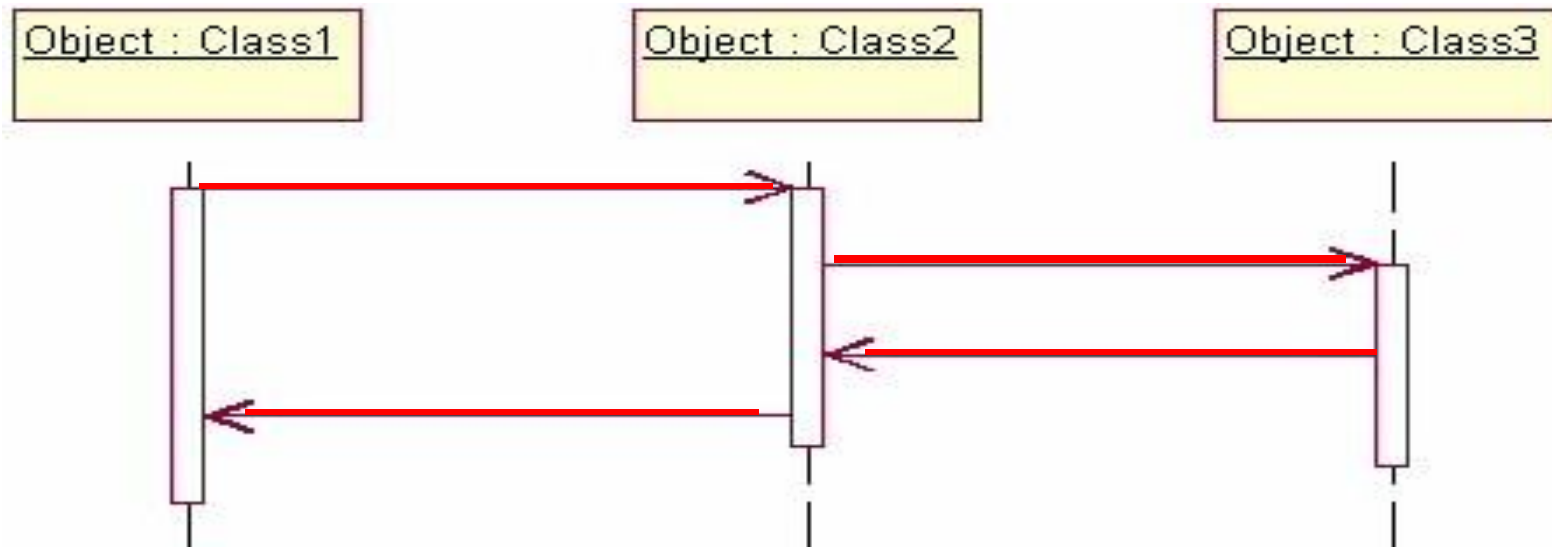




# Diagrama de Secuencia



# 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

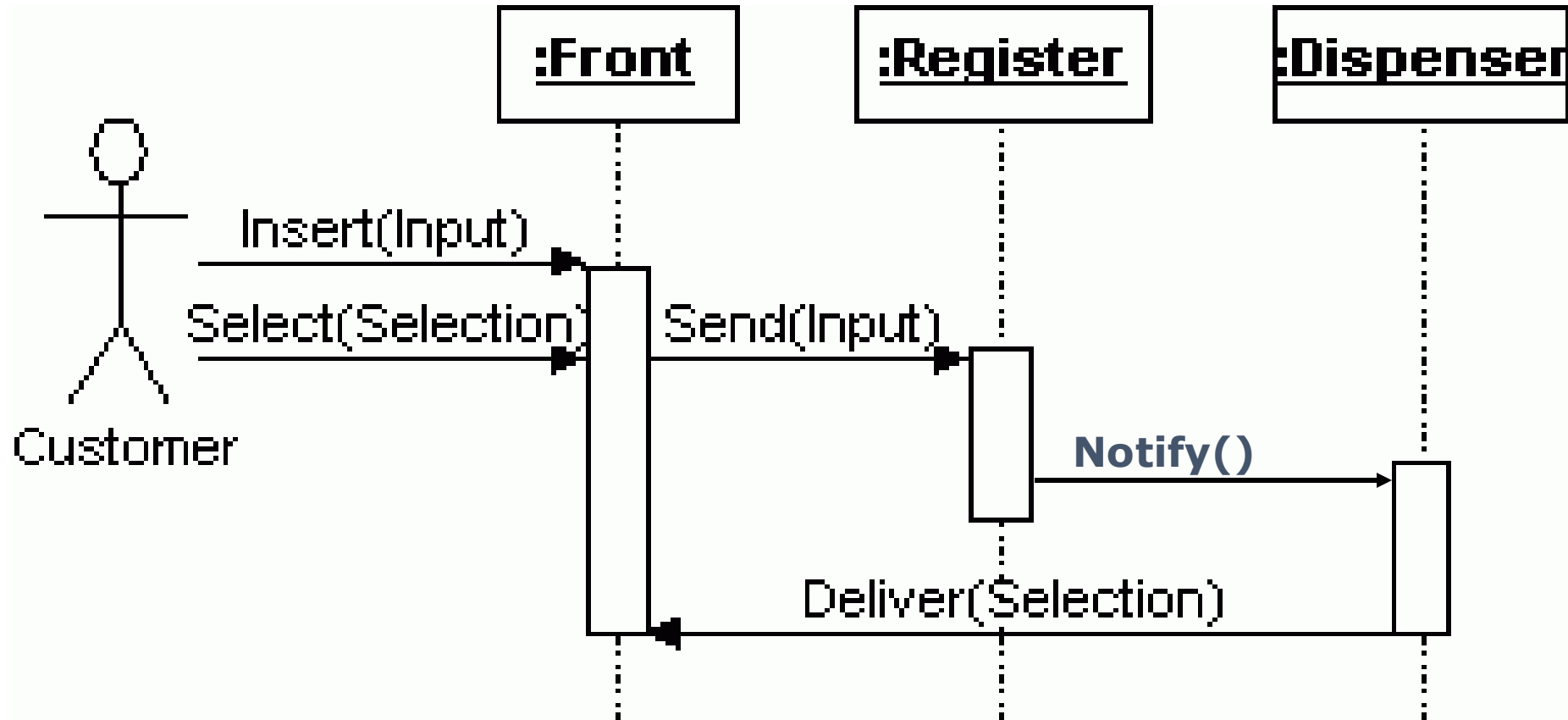


# Ejemplo

- 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



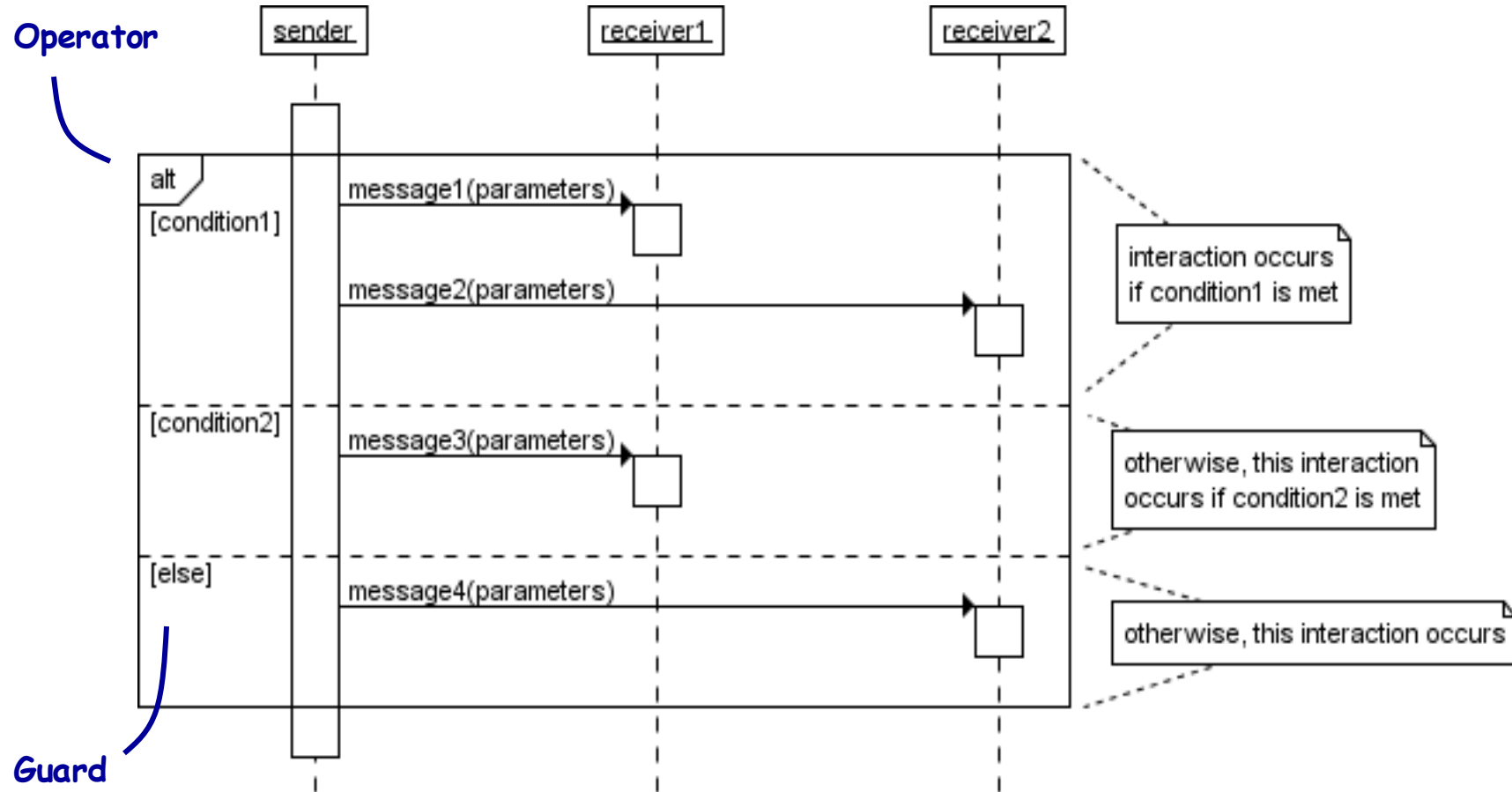
*The "Buy a product" scenario.*

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

## ... Pero

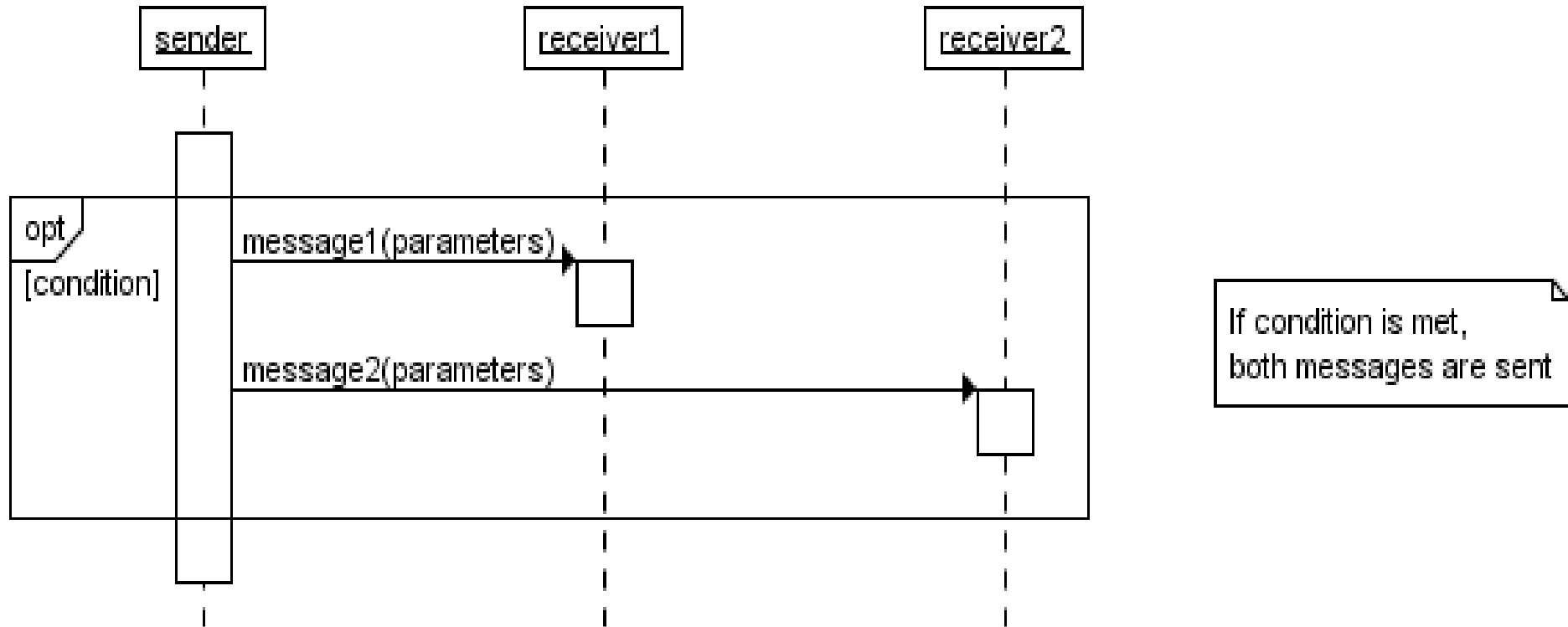
- Hemos visto una instancia de un diagrama de interacción: una posible secuencia de mensajes
- Dado que un caso de uso puede incluir muchos escenarios
  - Es necesario mostrar un comportamiento condicional.
  - Es necesario mostrar posibles iteraciones
- Un diagrama de interacción genérico muestra todas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

# alt: Operadores en frames de interacciones – UML 2.0



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

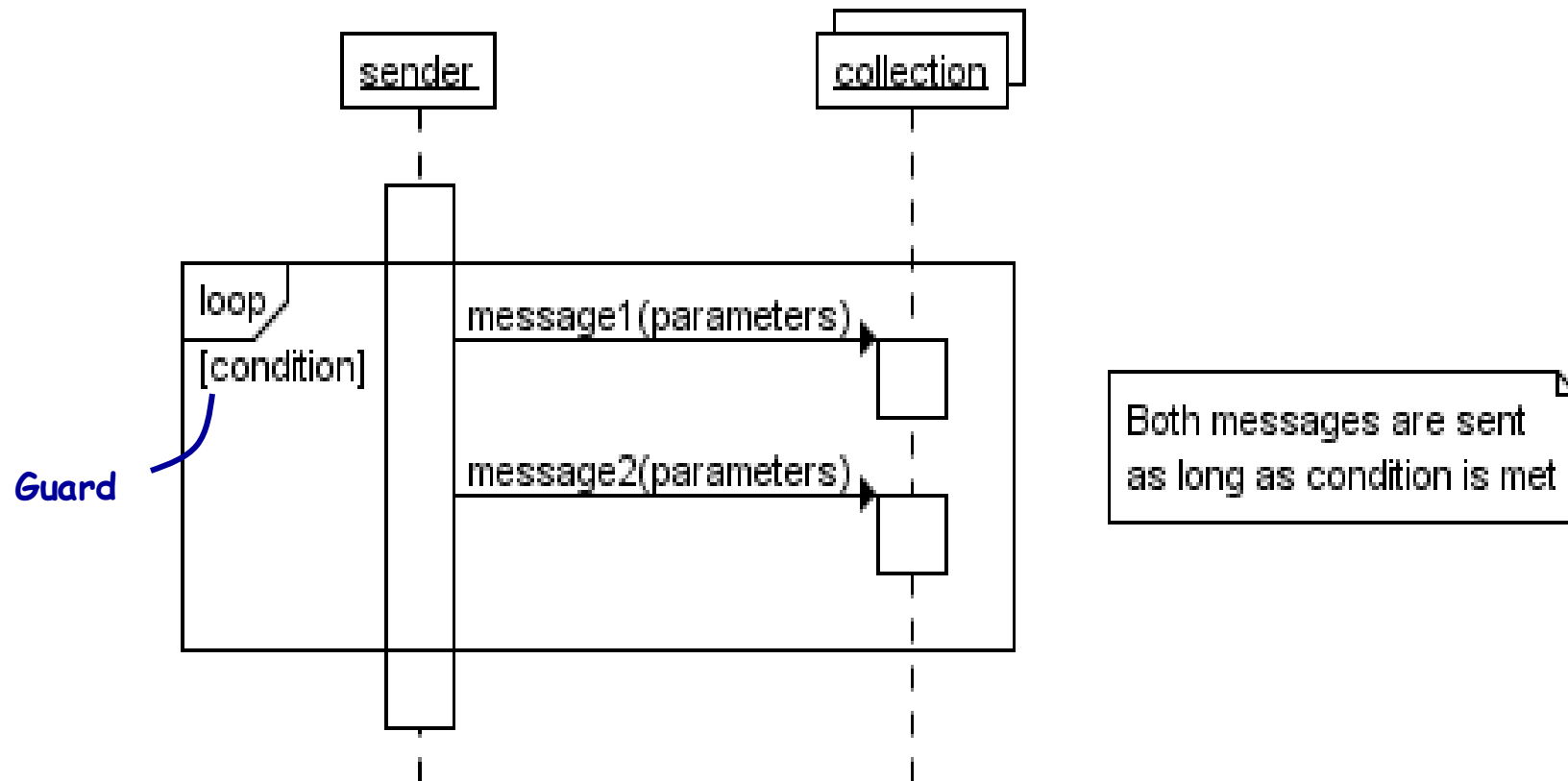
# Opt en UML 2.0



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

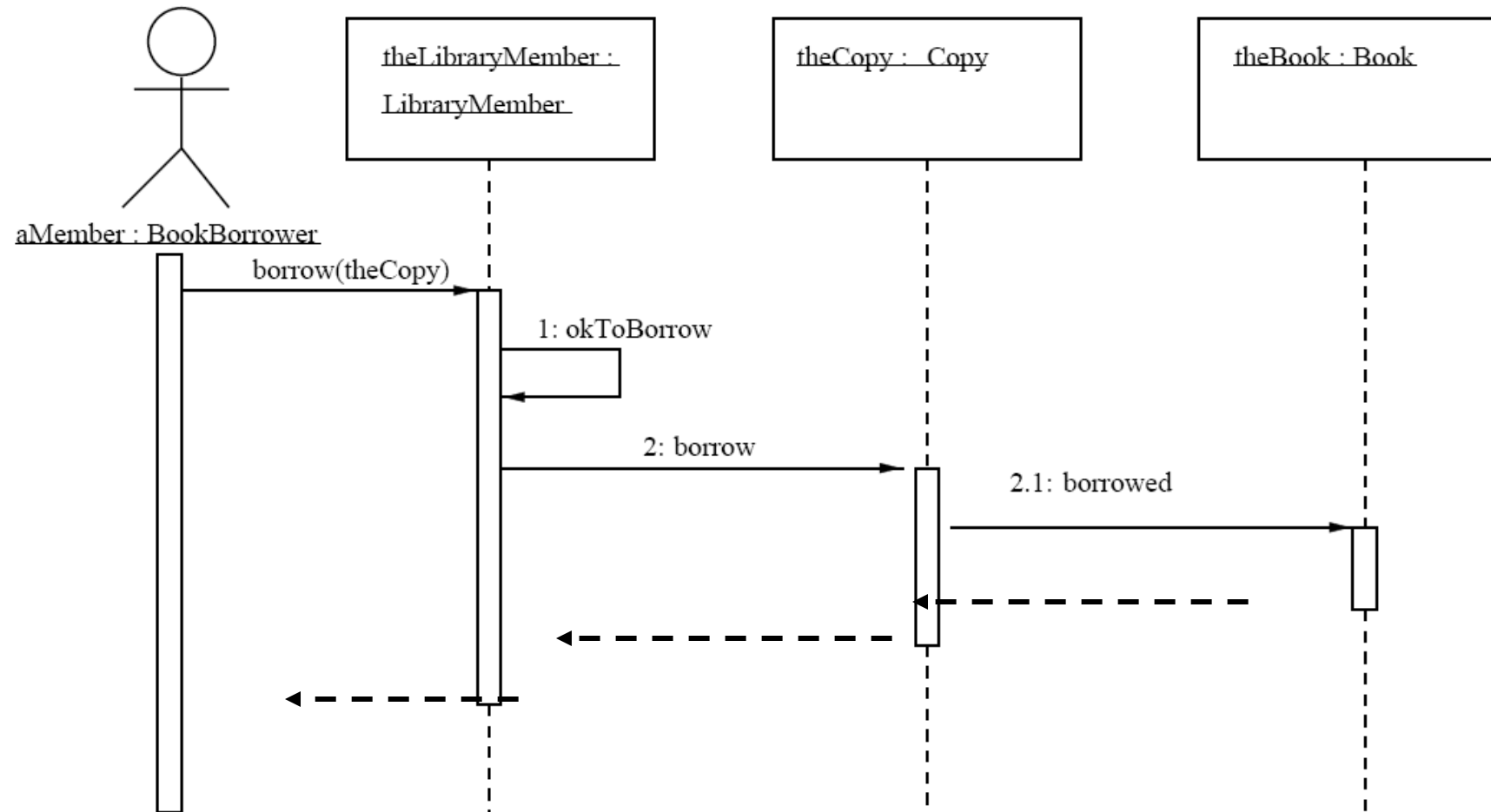


# Loop en UML 2.0

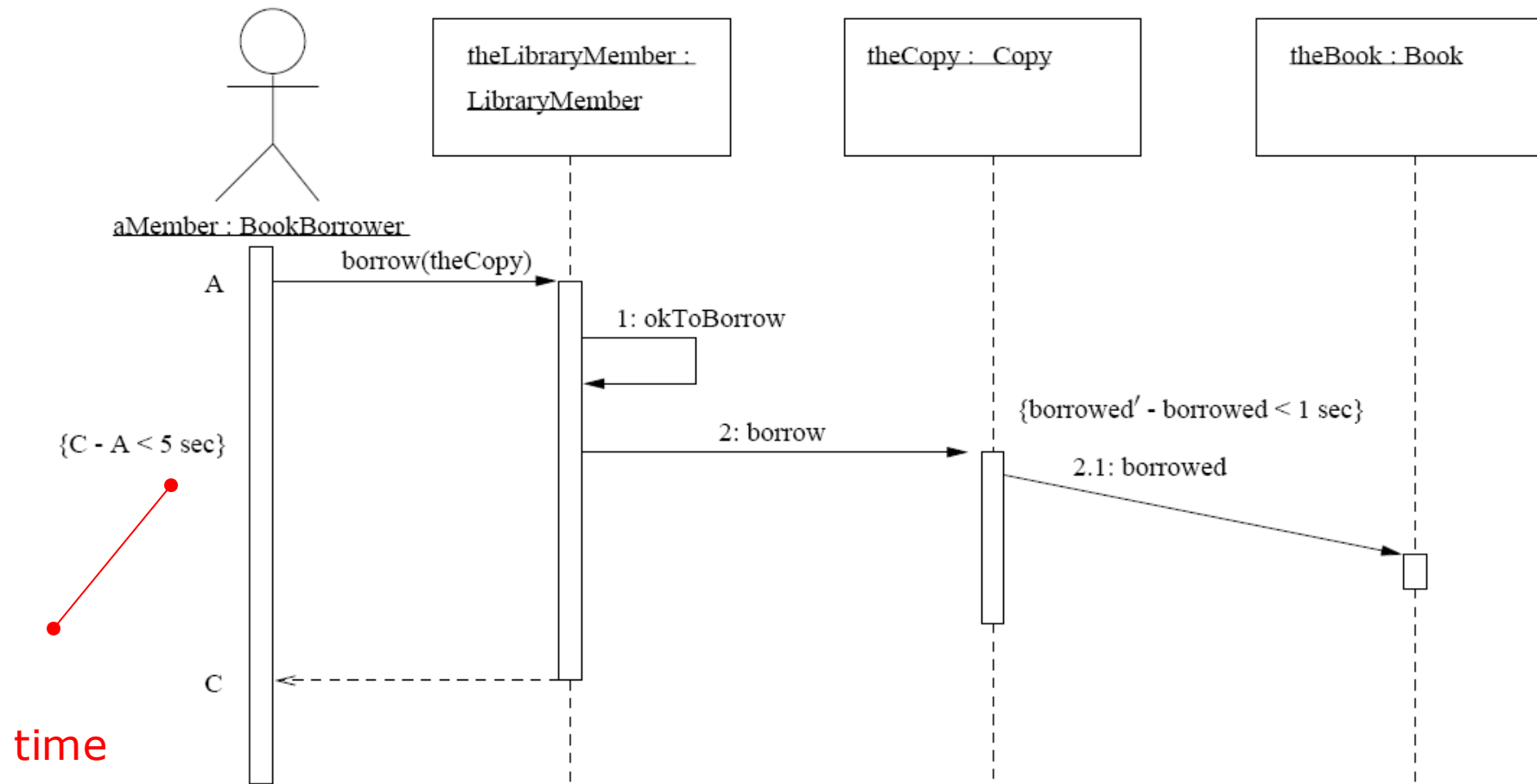


**Loop: the fragment may execute multiple times, and the guard indicates basis for iterations**

# Diagrama de Secuencia de la Biblioteca



# Mostrando restricciones de tiempo en un diagrama



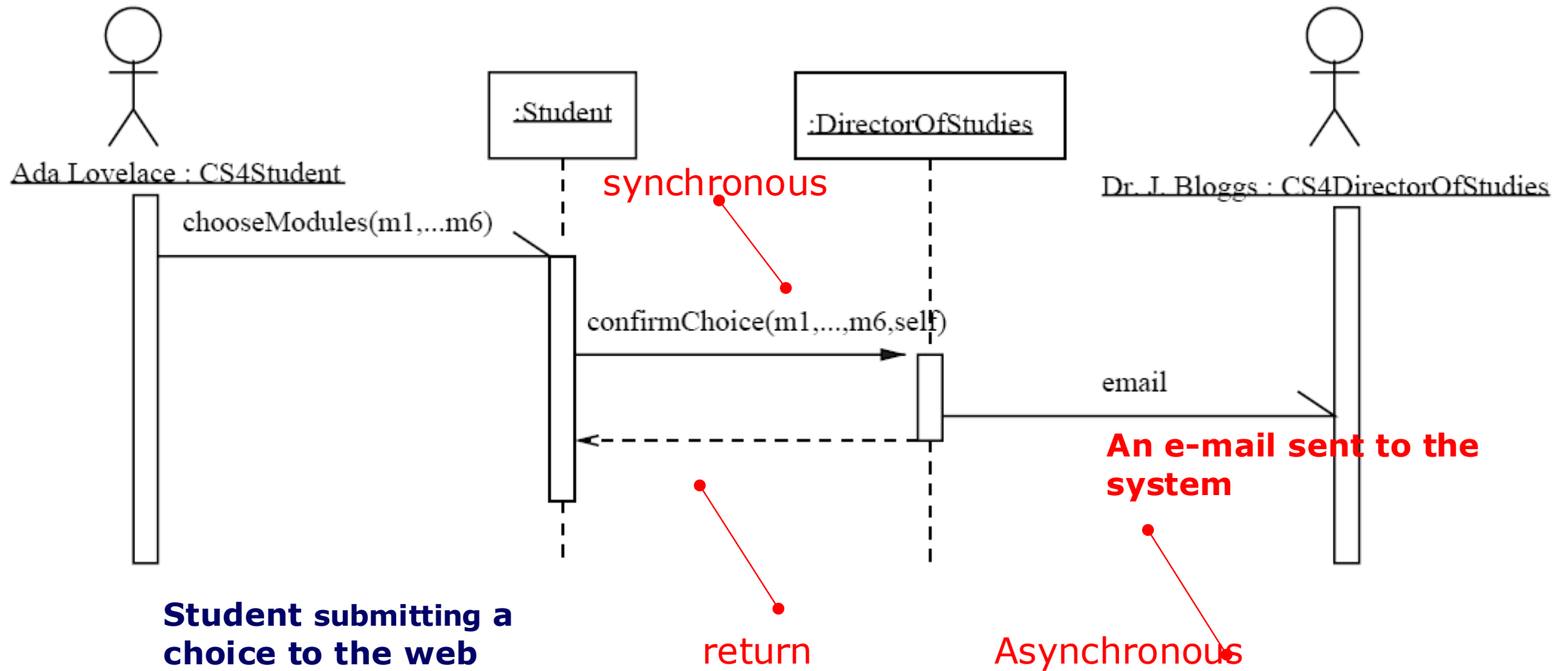
# Tipos de Interacciones en Diagramas de Secuencia

Some UML  
versions use  
for both

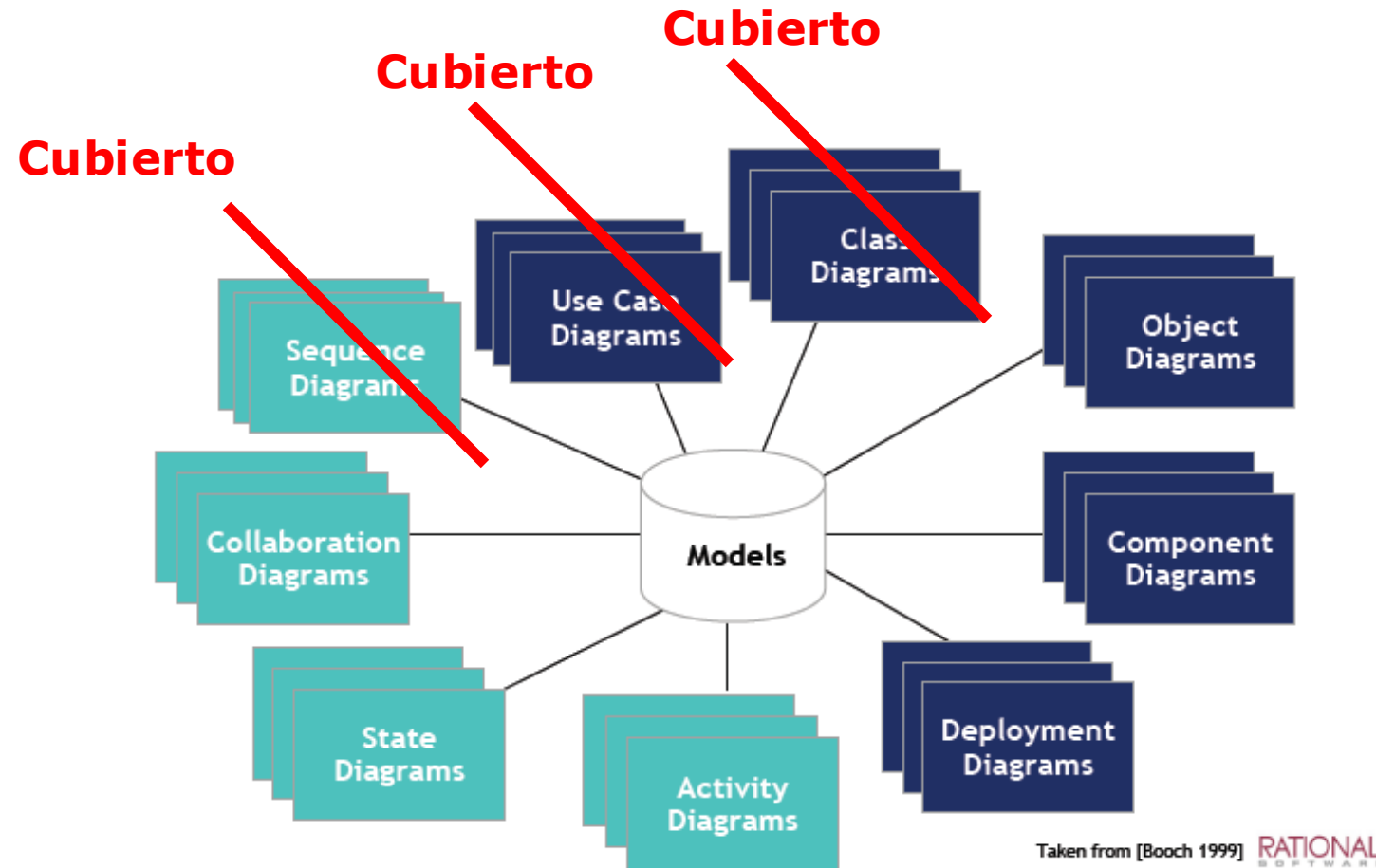


Interaction type	Symbol	Meaning
Synchronous or call	→	The 'normal' procedural situation. The sender loses control until the receiver finishes handling the message, then gets control back, which can optionally be shown as a return arrow.
Return	←	Not a message, but a return from an earlier message. Unblocks 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



# 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.