

# Introduzione a UML

## UML: Casi d'Uso e narrative

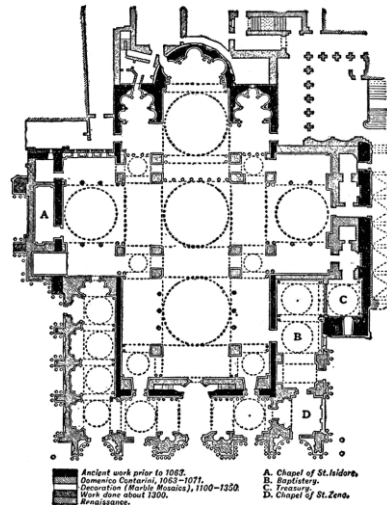
Laura Semini  
Ingegneria del Software  
Dipartimento di Informatica  
Università di Pisa

# Outline della lezione

- Questa lezione
  - Introduzione a UML
  - Diagramma dei casi d'uso

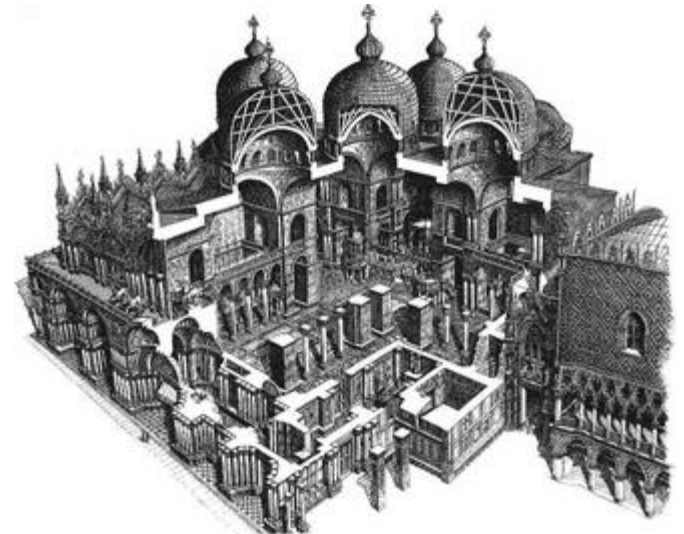
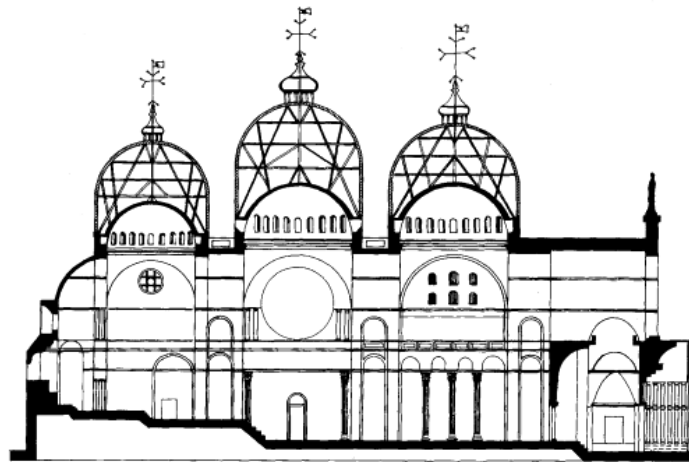
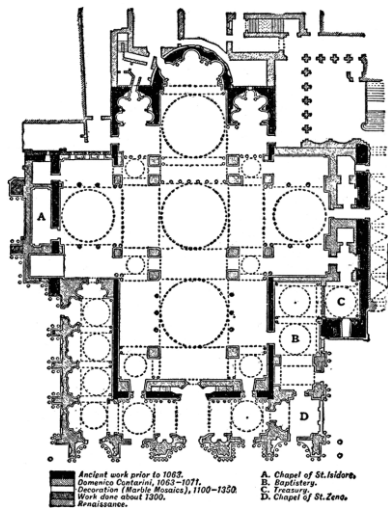
# Cosa è un modello

- Astrazione del sistema (o del dominio) usato per specificarne struttura e/o comportamento
- Un modello:
  - cattura aspetti importanti e astrae dai dettagli
  - è espresso con un formalismo che ne rende facile l'uso e la comprensione



# Diverse viste su un modello

- Un sistema è descritto non da una singola vista ma da un insieme di viste che nel loro insieme ci danno una visione completa del modello



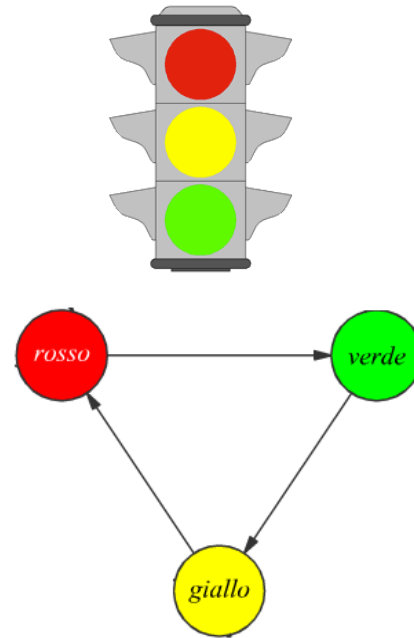
# Ancora sui modelli

- Strumento di documentazione, comunicazione e discussione
  - Fondamentale per un processo di progettazione collaborativo, al centro del moderno sviluppo sw
- 1. Come si modella un sistema?
- 2. Come si rappresenta un modello?
- 3. Come si usa un modello?

# 1. Come si modella un sistema?

Si decide se fornire:

- modello statico
- modello dinamico
- entrambi



In ogni caso si decide a che livello di astrazione porsi

## 2. Come si rappresenta un modello?

- Con un linguaggio
  - formale o
  - Semi-formale
- Adatto a descrivere aspetti diversi del progetto:
  - Descrizione strutturale (statica) vs descrizione di un comportamento (dinamica)
  - Bozza di progetto vs progetto finale
  - Descrizione del dominio vs descrizione della struttura del codice

# 3. Come si usa un modello?

- Un modello può essere:
  - Bozza (sketch)
    - *modello non completo, usato per descrizioni iniziali*
    - *Si possono rendere visibili soluzioni alternative*
  - Progetto dettagliato (blueprint)
    - *contiene dettagli sufficienti per permettere agli sviluppatori di creare un ready-to-run system senza prendere decisioni di progettazione*
  - Eseguibile
    - *talmente completo e preciso da permettere di generare il codice automaticamente dal modello*



# UML

## Unified Modeling Language: linguaggio di modellazione unificato

- Famiglia di notazioni grafiche
  - Permettono di descrivere diversi modelli da diversi punti di vista
  - Descritte da un meta-modello comune
  - Sono generalmente facilmente comprensibili a clienti e utenti
- Serve per supportare la descrizione e il progetto di sistemi software in particolare applicazioni sw Object Oriented, ma non solo....

# Breve cronistoria

- . . . 1994: molti linguaggi e metodi di modellazione OO. Esistevano comunque alcuni primi della classe - Booch e Rumbaugh per i linguaggi e Jacobson per le metodologie
- 1994: primo tentativo di unificazione - il linguaggio Fusion
- 1994: Booch e Rumbaugh si uniscono nella Rational Corporation dando vita a quello che chiamano UML
- 1996: Object Management Group (OMG) lancia una prima proposta di standardizzazione di UML
- 1997: OMG approva lo standard UML 1.0
- 2000: UML 1.4
- 2006: UML 2.0
- 2006 . . . : Model Driven Architecture (MDA)
- Attualmente UML 2.5.1

# Generalità

- Unificazione a livello di linguaggio
- Modellazione di ogni fase del processo di sviluppo
- Applicabile a più tipi di progetti e domini
- Indipendente
  - dal linguaggio di sviluppo
  - dal modello di ciclo di vita (UP consigliato, ma non imposto)

# Obiettivi

- Visualizzazione
  - Comunicazione e comprensione
- Specifica e documentazione
  - Descrizione del sistema in tutti i suoi livelli
- Realizzazione
  - Supporto all'automazione della codifica

# Modello statico e dinamico

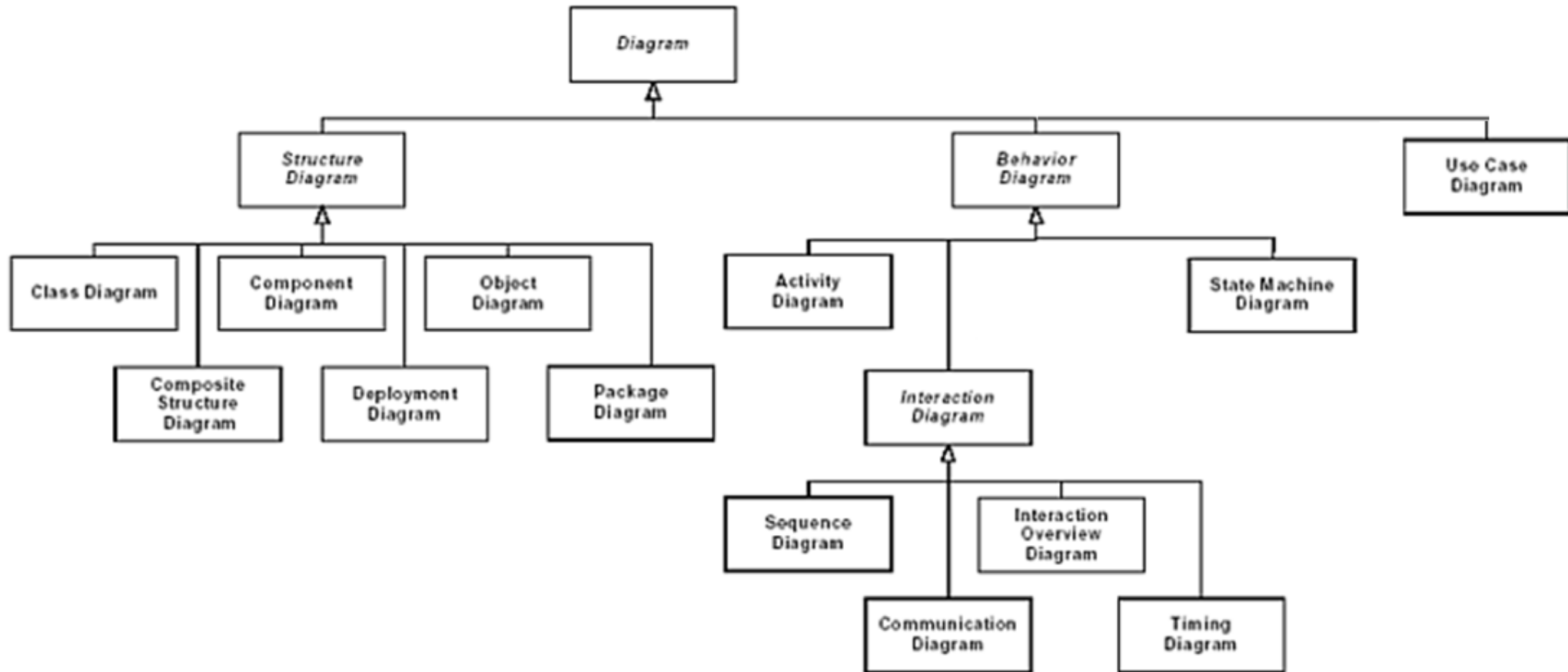
## Modello statico

- Entità e relazioni per descrivere:
  - Concetti del dominio
  - Componenti architettura
  - Classi di realizzazione
- Assenza degli aspetti dipendenti dal tempo

## Modello Dinamico

- Modella il comportamento delle entità descritte nel modello statico

# Diagrammi in UML: 14 diversi



# Diagrammi che vedremo nel corso

- **Diag. dei casi d'uso**
- **Modello statico**
  - Diag. delle classi
  - Diag. degli oggetti
  - Diag. delle componenti
  - Diag. di struttura composita
  - Diag. di dislocazione
  - Diag. dei package
- **Modello dinamico**
  - Diag. di macchina a stati
  - Diag. di attività
  - Diag. di sequenza

Diagramma:

- **Rappresentazione grafica**
  - di un insieme di elementi del modello
- **Grafo**
  - nodi = elementi del modello
  - archi = relazioni fra elementi

# Diagramma dei casi d'uso

---



# Contenuti

- Modellazione dei Requisiti:
  - aiuta a raccogliere i requisiti, elaborarli e documentarli
- Modello statico:
  - diagramma dei casi d'uso
- Modello dinamico:
  - narrazioni associate ai casi d'uso

# Diagramma dei casi d'uso

- Descrive i requisiti funzionali del sistema
- Cattura le funzionalità che un sistema deve offrire, visto dall'esterno:  
*i compiti che un utente può fare con l'aiuto del sistema.*
- Un *attore* è un'entità esterna al sistema, che interagisce direttamente con esso in un determinato ruolo
  - utente
  - altro sistema
  - tempo (attore speciale)
- Un caso d'uso è
  - una funzionalità o un servizio offerto dal sistema a uno o più attori
  - formalmente: un compito che un attore può svolgere con l'aiuto del sistema
  - espressa come un insieme di SCENARI
- Uno scenario è
  - Una sequenza di interazioni (scambi di messaggi) tra sistema e attori

# Costruzione del diagramma dei casi d'uso

La modellazione dei requisiti usando il diagramma dei casi d'uso prevede i seguenti passi:

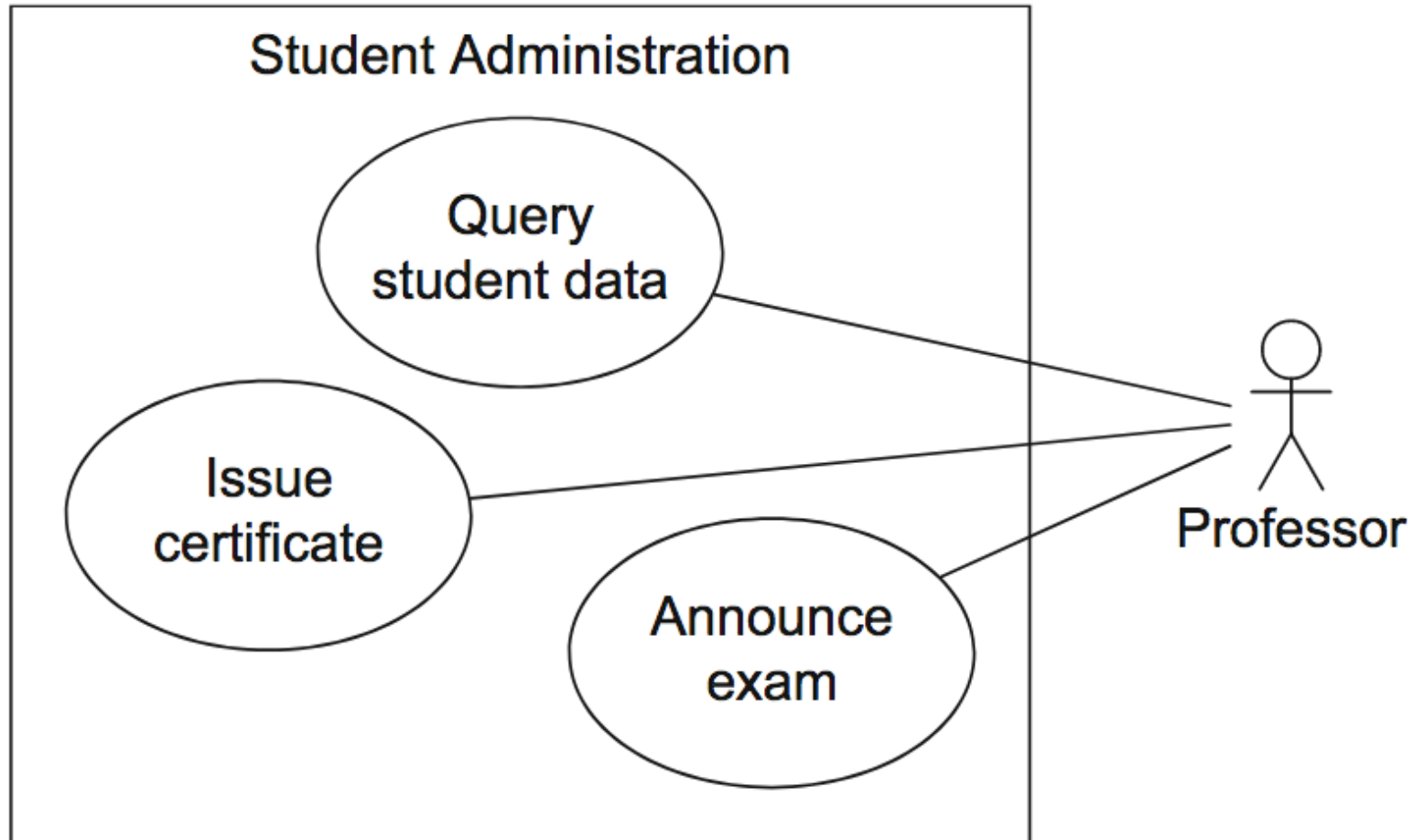
- Individuare il confine del sistema
- Individuare gli attori
- Individuare i casi d'uso
- Individuare le relazioni attore-caso d'uso
  
- Specificare il caso d'uso
  - con una descrizione testuale (narrativa)

# Diagramma dei casi d'uso

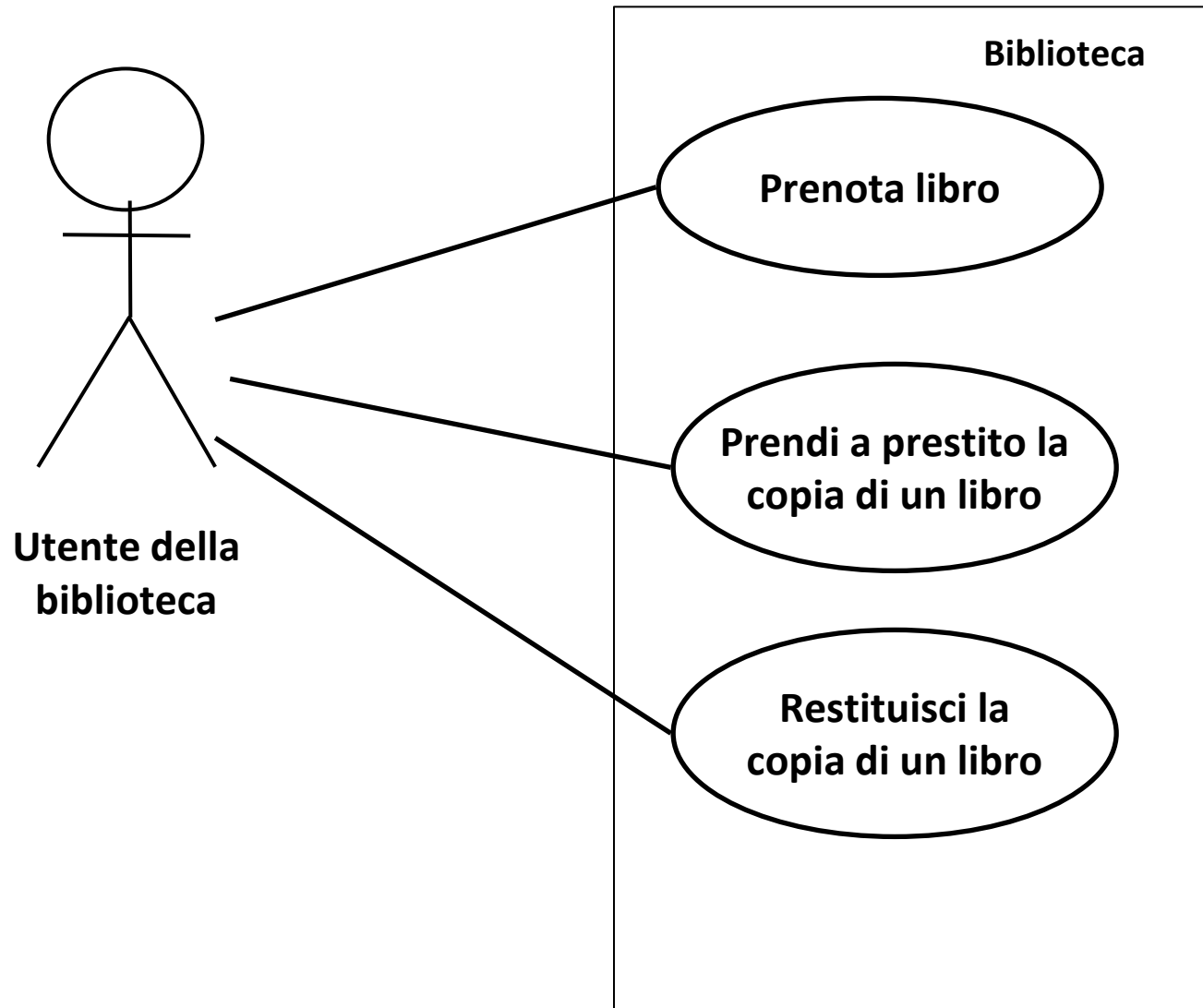
Il diagramma è composto da:

- **attori**: i ruoli assunti dalle persone e dalle cose (macchine) che usano il sistema
- **casi d'uso**: quello che gli attori possono fare con il sistema
- **relazioni**: relazioni significative tra gli attori e casi d'uso
- **confine del sistema**: un rettangolo disegnato intorno ai casi d'uso per indicare il confine del sistema oggetto del modello

# Primo esempio di diagramma dei casi d'uso



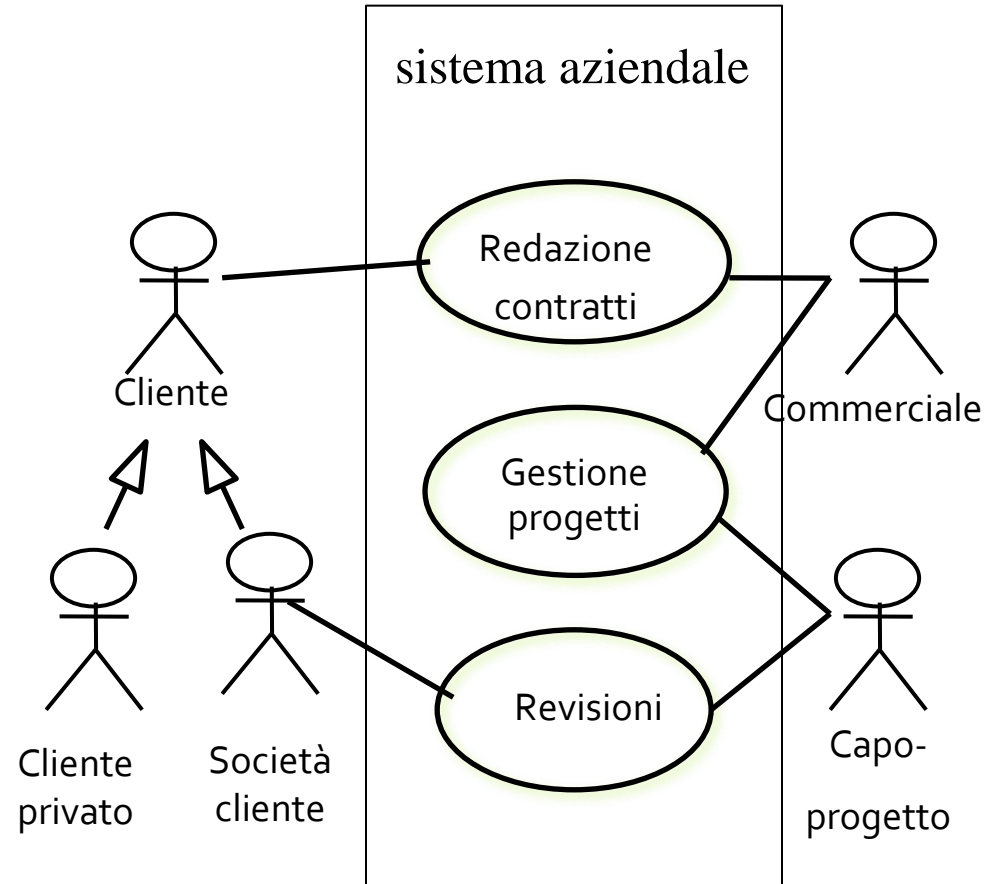
# Secondo esempio di diagramma dei casi d'uso



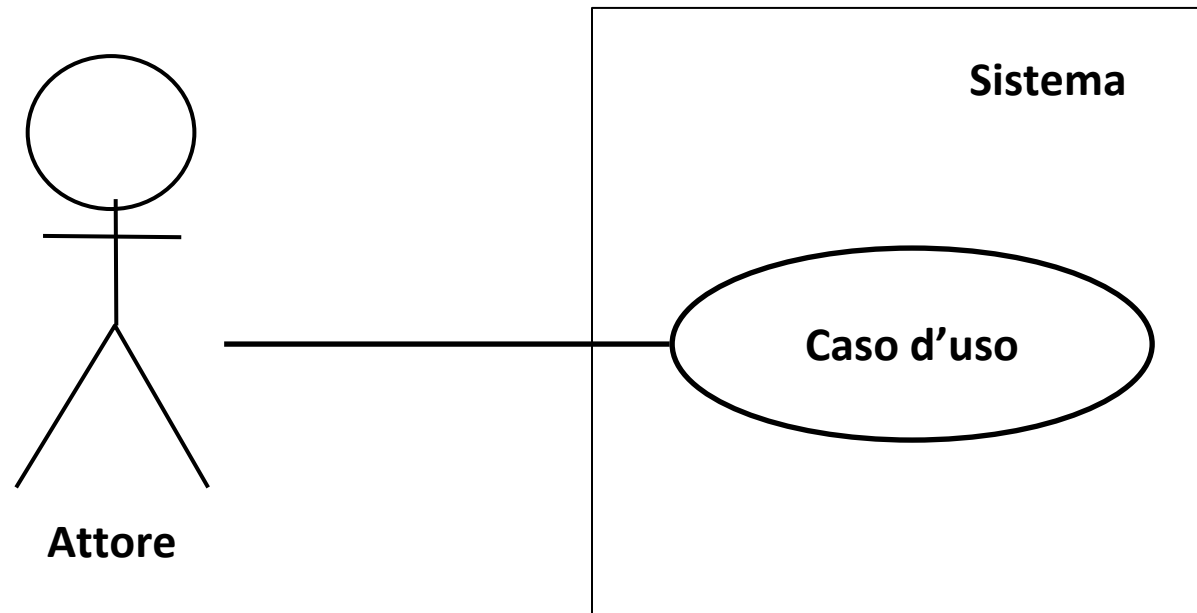
# Terzo esempio di diagramma dei casi d'uso

## ■ Esempio

- I clienti possono essere privati o società
- Clienti e commerciali partecipano alla redazione dei contratti
- Società e capiprogetto partecipano alle revisioni
- Commerciali e capiprogetto gestiscono i progetti



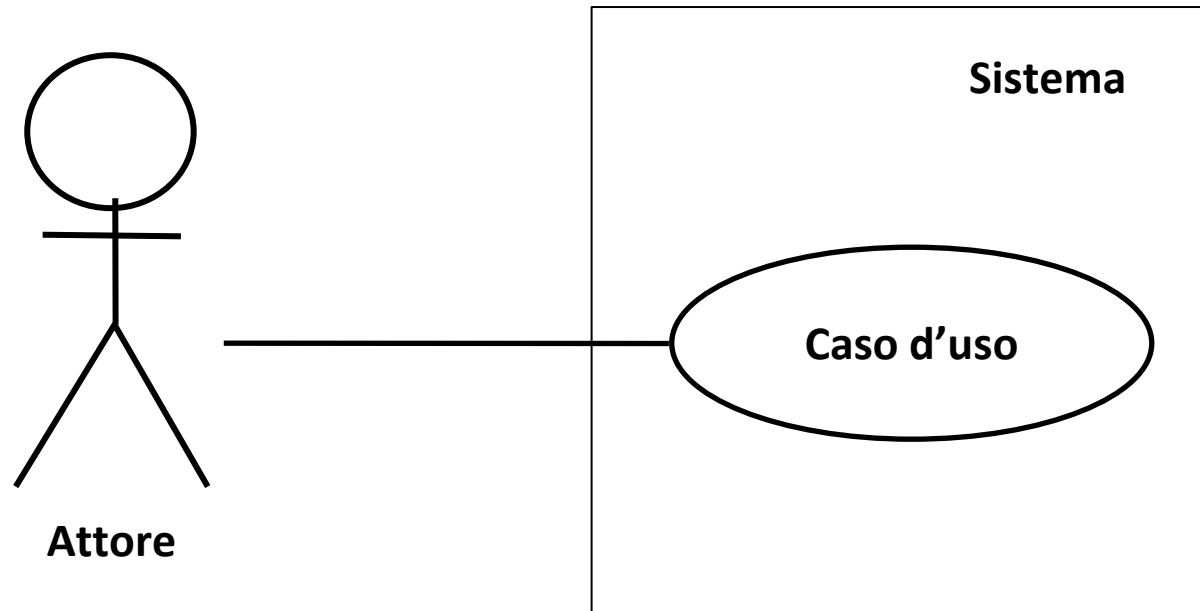
# Sintassi del caso d'uso



- Attore: omino con nome (Maiuscolo, volendo UpperCamelCase) (è una classe)
- Associazione: senza nome
- Caso d'uso: ovale con nome (Maiuscolo, volendo UpperCamelCase) (verbo)
- Sistema: rettangolo con o senza nome, contiene i casi d'uso
- L'associazione attori—casi d'uso è molti a molti: un attore può essere associato a più casi d'uso e viceversa
- Casi d'uso senza attori solo se: inclusi (included), estensioni o sotto-casi



# Semantica del caso d'uso



- Un attore è un utente o un altro sistema, in un particolare ruolo
- Un caso d'uso è un compito (task) che gli attori eseguono con l'ausilio del sistema
- L'associazione attori—casi rappresenta un'interazione (sequenza di messaggi).
- Un caso d'uso è iniziato SOLO da un attore (principale: la narrativa definisce che è l'attore principale)
  - **Eventualmente Tempo**

# Casi d'uso: descrizione narrativa

- Documento che descrive il modello dinamico
  - gli scenari rilevanti di un caso d'uso
  - dal punto di vista degli attori
- Caratteristiche
  - Inizio, fine, passi intermedi
  - Condizioni eccezionali
  - Precondizioni, postcondizioni

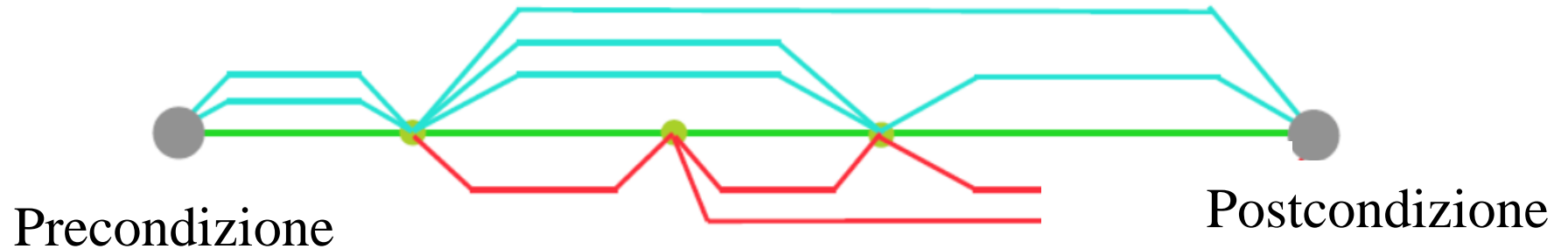
# Descrizione narrativa

<i>Nome:</i> <i>ID:</i>	Nome del caso d'uso identificatore
<i>Breve descrizione:</i>	Due righe riassuntive
<i>Attori primari:</i>	Attori che avviano il caso d'uso
<i>Attori secondari:</i>	Altri attori che interagiscono con il caso d'uso
<i>Precondizioni:</i>	Devono valere prima dell'esecuzione del caso d'uso
<i>Sequenza degli eventi principale:</i>	Sequenza di passi
<i>Postcondizioni:</i>	Della sequenza principale
<i>Sequenze alternative degli eventi:</i>	Errori, ramificazioni e interruzioni nella sequenza principale

# Esempio di descrizione narrativa

<i>Nome:</i>	Estendi il prestito
<i>Breve descrizione:</i>	Un utente desidera prolungare il periodo di prestito
<i>Attori primari:</i>	Utente della biblioteca
<i>Attori secondari:</i>	Nessuno
<i>Precondizioni:</i>	Il libro è già assegnato all'utente
<i>Sequenza degli eventi principale:</i>	<ol style="list-style-type: none"><li>1. L'utente chiede di prolungare il prestito</li><li>2. Il Sistema accerta che nessuno l'abbia prenotato</li><li>3. Il Sistema accerta che l'utente sia ammesso al prestito</li><li>4. Il Sistema aggiorna utente e stato del libro</li></ol>
<i>Postcondizioni:</i>	Prestito esteso, stato utente aggiornato
<i>Sequenze alternative degli eventi:</i>	Il libro è già prenotato, utente non ammesso al prestito

# Casi d'uso e scenari



- Uno scenario è un'istanza di un caso d'uso: una sequenza di passi che produce un risultato osservabile da uno o più attori
- Gli scenari descritti dalla sequenza principale degli eventi sono quelli che portano alla postcondizione

# Relazione tra preconditione, postcondizione, sequenza principale e sequenza alternativa

- Parallelo con le triple di Hoare:
- {Precondizione} sequenza principale {Postcondizione}
- Quindi:
  - per ogni stato  $\sigma$  che soddisfa la preconditione, l'esecuzione del caso d'uso a partire da  $\sigma$  termina producendo uno stato  $\sigma'$  che soddisfa la postcondizione
- ... a meno che non si verifichi qualche imprevisto come quelli elencati nella sequenza alternativa, in questo caso la postcondizione non è garantita

# Sintassi della sequenza principale degli eventi (S.P.E.)

- La S.P.E. elenca i passi che compongono il caso d'uso.
- La sequenza è numerata e ogni passo dovrebbe avere la struttura:  
*<numero>. <soggetto><azione><complementi>*
- Il primo passo (attivazione) è sempre compiuto dall'attore principale.
  1. L'utente chiede di prolungare il prestito
  2. Il sistema controlla .....
  3. ....

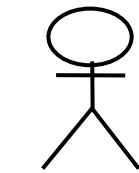
# Sintassi della sequenza principale degli eventi (S.P.E.)

Nella S.P.E. si possono descrivere condizioni e cicli. In questo caso conviene identificare i passi interni con indici annidati

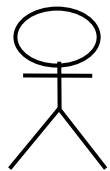
- x. Se (espressione booleana)
  - x.1 ....
  - x.2 ....
  - Altrimenti [OPZIONALE]
  - x.3 ....
- y. Per (espressione di iterazione)
  - y.1 ....
- z. While (espressione booleana)
  - z.1 ...
  - z.2 ...



# Casi d'uso: generalizzazione tra attori



Cliente



Cliente privato



{abstract}  
Research  
Associate



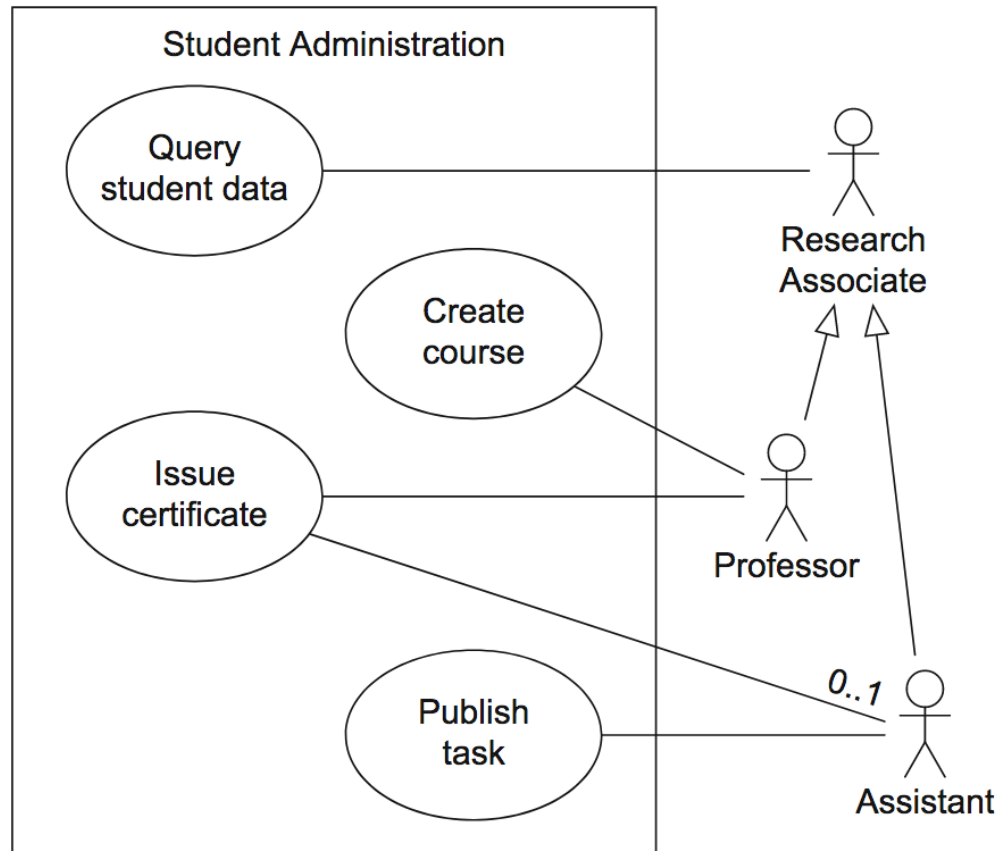
Professor



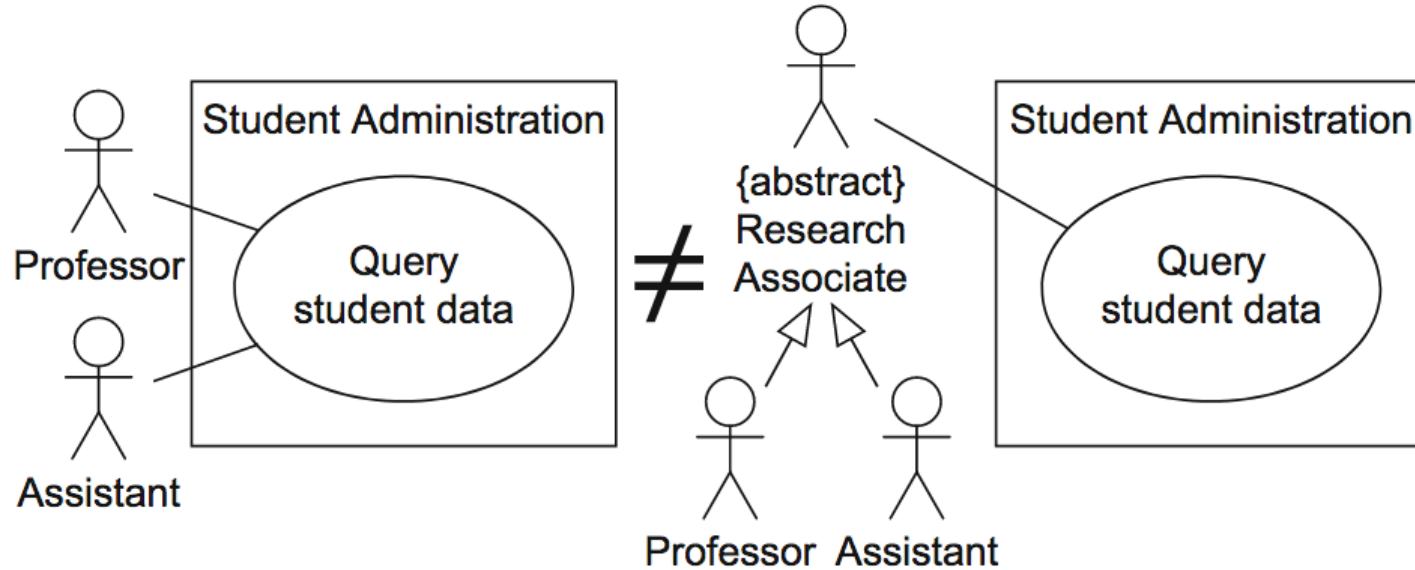
Assistant

È possibile che  
l'attore padre  
sia astratto

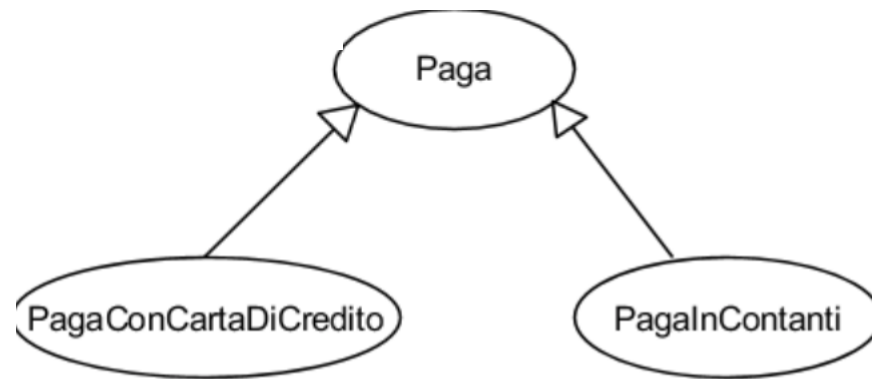
# Esempi di generalizzazione



# Esempi di generalizzazione

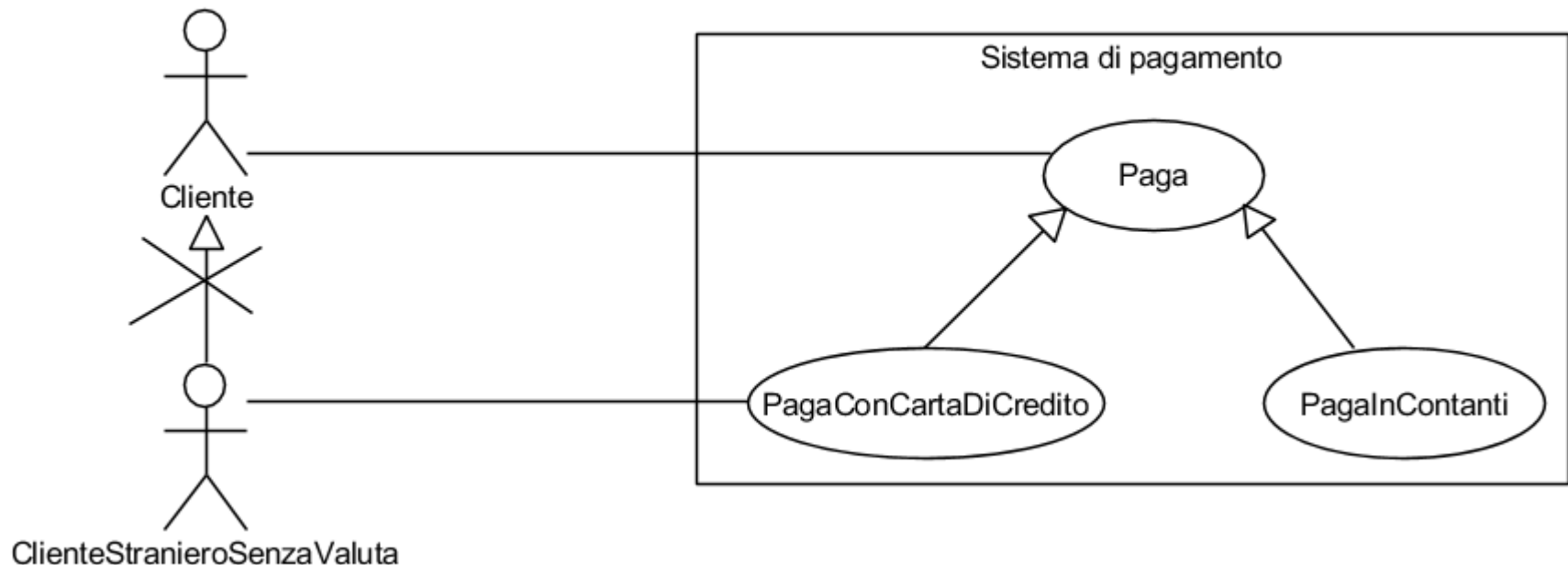


# Generalizzazione dei casi d'uso

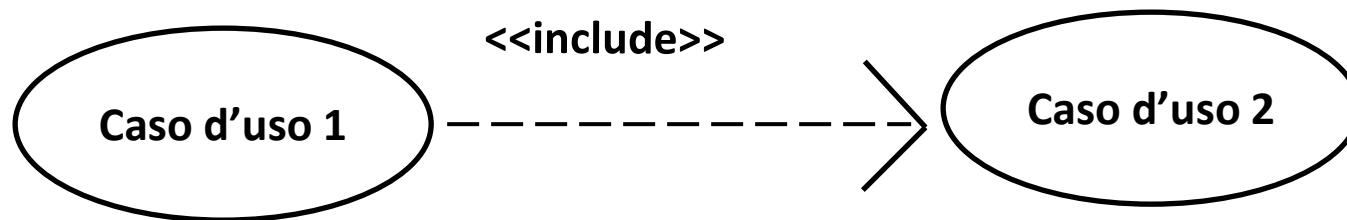


# Attenti alla generalizzazione...

Usando la generalizzazione fare attenzione che il classificatore specializzato eredita tutte le relazioni del classificatore padre (Liskov)

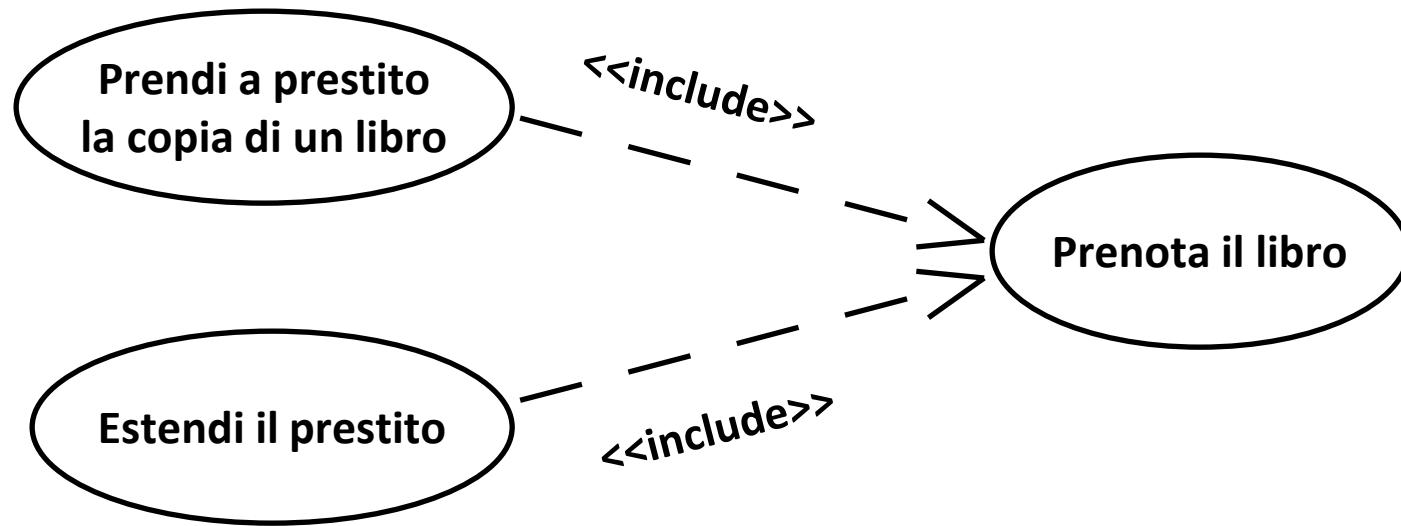


# Inclusione di caso d'uso



- La relazione include punta al caso d'uso che si include
  - Il caso d'uso 1 incorpora l'interazione descritta dal caso d'uso 2 come parte dell'interazione che esso (1) descrive
- Simile ad una chiamata di funzione
  - La narrazione del caso 1 dovrebbe invocare l'esecuzione del caso incluso (2)
- <<include>> è uno stereotipo
  - Gli stereotipi sono keywords che annotano elementi di un diagramma, per precisarne il significato
  - In questo caso specializza la relazione di dipendenza ----> dicendo che si tratta di un'inclusione

# Esempio di inclusione



Sia per prendere in prestito la copia di un libro che per estendere il prestito occorre prenotare il libro

# Sequenza degli eventi principale: include, caso base

1. ...

2. ...

3. Include (nome caso incluso)

4. ...

■ Il caso d'uso incluso è specificato a parte

- Può essere istanziabile (completo) (avviato da un attore)
- ....o non istanziabile (eseguito solo quando incluso)



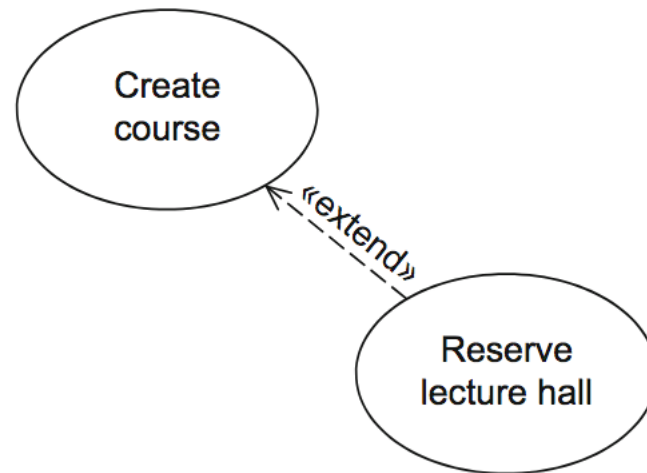
# Estensione di un caso d'uso



- La relazione extend punta al caso d'uso che viene esteso
  - Il caso d'uso 1 puo incorporare l'interazione descritta dal caso d'uso 2
  - La narrazione del caso 1 puo' in qualche caso invocare l'esecuzione del caso (2)
- <<extend>> è uno stereotipo
  - Gli stereotipi sono keywords che annotano elementi di un diagramma, per precisarne il significato
  - In questo caso specializza la relazione di dipendenza ----> dicendo che si tratta di una estensione

Attenzione al verso della freccia  
opposto all'include

# Esempio di Estensione



Quando si crea un nuovo corso **si può** riservare un'aula ma **non è obbligatorio**

# Syllabus

UML@Classroom, Martina Seidl, Marion Scholz,  
Christian Huemer, Gerti Kappel, Springer Verlag, 2015

Cap 1-2-3

# "Per casa": Chiavi magnetiche

- Disegnare il diagramma dei casi d'uso del seguente sistema
- Per motivi di sicurezza, un'organizzazione ha deciso di realizzare un sistema secondo il quale a ogni dipendente è assegnata una chiave magnetica per accedere (aprire) determinate stanze. I diritti di accesso dipenderanno in generale dalla posizione e dalle responsabilità del dipendente. Quindi sono necessarie operazioni per modificare i diritti di accesso posseduti da una chiave se il suo proprietario cambia ruolo nell'organizzazione.