

# Introduzione a UML

## UML: Casi d'Uso e narrative

Vincenzo Gervasi, Laura Semini  
Ingegneria del Software  
Dipartimento di Informatica  
Università di Pisa

# Riassunto lezione precedente

## Outline della lezione

- Lezione precedente:
  - Il metodo Jackson
- Questa lezione
  - Introduzione a UML
  - Diagramma dei casi d'uso

# Attività di analisi e progettazione Modelli

- Necessità di un maggior supporto alle attività di analisi e progettazione
- Strumenti automatici di supporto alla realizzazione
- Realizzazione
  - Centrata sul codice
  - Verifica del codice rispetto alla specifica
  - Revisione del codice per risolvere i problemi
- Analisi e progettazione
  - Centrate sul modello
  - Verifica del modello rispetto ai requisiti
  - Revisione del modello per risolvere i problemi

# Cosa è un modello

- Astrazioni del sistema (o del dominio) usati per specificarne struttura e comportamento
  - Auto: motore, sterzo, ruote, acceleratore, freno
  - Elettromagnetismo: equazioni di Maxwell
- Diverso dal modello computazionale: il codice è un modello del calcolo eseguito dalla macchina

# Uso di un modello

- Contiene la conoscenza sul “cosa” e sul “come” di un sistema
  - Strumento di comunicazione e discussione
  - Documentazione
- Fondamentale per un processo di progettazione collaborativo, al centro del moderno sviluppo sw
- Come si modella un sistema?
- Come si rappresenta un modello?

# UML e UP

- Unified Modeling Language: linguaggio di modellazione grafico di sistemi software
  - Generale: non è esclusivamente un linguaggio per definire applicazioni sw OO
  - i diagrammi sono generalmente facilmente comprensibili agli utenti con un minimo di conoscenze nel campo
- Unified Process: specifica un processo di sviluppo del sw
  - indica le attività che devono essere eseguite, i manufatti etc etc
- UP descrive attività e “raccomanda” uso di UML. Vivono bene in simbiosi ma l’uno non implica l’altro

# 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 - linguaggi di azione
- 2006: UML 2.0
- 2006 . . . : Model Driven Architecture (MDA)

# Generalità

- Modellazione di ogni fase del processo di sviluppo
- Applicabile a più tipi di progetti e domini
- Indipendente
  - dal linguaggio di sviluppo
  - dal modello di processo
- Unificazione a livello di linguaggio, non di metodo!!



# Obiettivi

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

# SU UML

- Principio base di UML è che un sistema software possa essere visto come un insieme di oggetti che collaborano
- Vengono considerati due aspetti fondamentali del sistema
  - Struttura statica: oggetti necessari e relazioni tra questi
  - Comportamento dinamico: come gli oggetti collaborano per raggiungere lo scopo

# alcuni concetti di UML

- **Modello: astrazione di (parte di) un sistema**
  - **ESEMPIO: Auto**
    - il modellino in scala
    - il progetto della stessa
    - la formula per il calcolo dello spazio di frenata
- **Modello statico**
  - Descrive gli elementi del sistema e le loro relazioni
- **Modello dinamico:**
  - Descrive il comportamento del sistema nel tempo
- **Progetto (o disegno)**
  - Insieme dei modelli: le diverse dimensioni del sistema
    - modelli statici: DB
    - dinamici: sistemi di controllo, di calcolo
- **Vista: descrizione di un aspetto di un modello**

# Caratteristiche dei modelli

- Tolleranza a inconsistenze e incompletezza
  - Supporto al dialogo tra le parti interessate
- Meccanismi di strutturazione (package)
- Personalizzazione mediante stereotipi
- Strumenti di supporto disponibili sul mercato

# come usare UML

- Un progetto può essere realizzato come
  - Abbozzo (sketch)
  - Progetto dettagliato (blueprint)
  - Eseguibile (UML come linguaggio di programmazione)
- Secondo due prospettive
  - Software: elementi UML corrispondono a elementi sw
  - Concettuale: elementi UML corrispondono a elementi del dominio

# un recap sui linguaggi

- Linguaggio formale = sintassi + semantica
- Sintassi
  - Le regole attraverso le quali gli elementi del linguaggio (ad esempio le classi) sono raggruppati in espressioni (ad esempio i diagrammi)
- Semantica
  - Le regole che assegnano un significato alle espressioni sintattiche

# Meta-modello di UML

- Per specificare la sintassi e la semantica
- Gli utenti sono i costruttori dei tool e i modellatori esperti
- UML descrive il proprio meta-modello in UML (come alcuni compilatori sono utilizzati per compilare se stessi)
- Stile semi-formale che combina linguaggio naturale e formale
- Esiste una rappresentazione XML del metamodello che si chiama XMI (XML Metadata Interchange)

# Elementi base di UML

- Entità
  - classi, interfacce, componenti, casi d'uso, ...
- Relazioni
  - associazioni, generalizzazioni, dipendenze, ...
- Diagrammi
  - delle classi, dei casi d'uso, d'interazione, ...



# Tipi di modelli

## MODELLO STATICO

- I concetti del dominio
  - modellati mediante entità e relazioni
- Realizzazione del sistema
  - classi di realizzazione, componenti....
- Assenza degli aspetti dipendenti dal tempo

## MODELLO DINAMICO

- Modella il comportamento delle entità descritte nel modello statico

# Diagrammi

- ...delle classi
  - ...degli oggetti
  - ...delle componenti
  - ...di struttura composita
  - ...di macchina a stati
  - ...dei casi d'uso
  - ...di attività
  - ...di sequenza
  - ...di dislocazione
  - ...dei package
- Rappresentazione grafica
    - di un insieme di elementi del modello
    - secondo una certa vista
  - Grafo
    - Vertici = elementi del modello
    - Archi = relazioni fra elementi

# Diagrammi e modelli

- I diagrammi si usano per costruire i modelli
- I diagrammi non sono i modelli
  - Un modello è un grafo di elementi *semantici*
  - Un diagramma è un grafo di elementi *visuali* che rappresentano gli elementi semantici
- Negli strumenti di supporto
  - Un modello è la struttura sulla quale si opera
  - Un diagramma è una struttura di presentazione
  - Non eliminare un elemento solo da un diagramma

# Classificatori vs istanze

- Un classificatore modella un concetto che descrive istanze
  - Ex. una classe modella oggetti
- I classificatori in UML sono
  - Classi
  - Attori
  - Casi d'uso
  - Componenti
  - Nodi (ex hw)...
- Alcuni diagrammi possono essere a livello di classificatore o a livello di istanza

# Annotazioni: commenti e vincoli

- Alcuni dettagli del modello si esprimono in forma testuale
- Commenti e vincoli sono annotazioni nei diagrammi
  - In OCL (Object Constraint Language), il linguaggio di vincoli di UML
  - In linguaggio naturale
- Attaccati ad un elemento del modello (o a una lista di elementi), con linea tratteggiata
- I vincoli tra parentesi graffe, i commenti senza graffe
- I vincoli sono elementi del modello, i commenti no



Questa è una nota



{Questo è un vincolo}

# Stereotipi

- Primitiva di UML comune ad ogni diagramma.
  - Rende un diagramma più informativo arricchendo la semantica dei costrutti UML.
- Uno stereotipo è una parola chiave tra virgolette e abbinata ad un elemento del modello.
  - Es. «import», «utility», «interface»

# Stereotipi cont'd

- Forniscono significato aggiuntivo ai costrutti UML.
- Possono essere usati per adattare UML a particolari ambiti e piattaforme di sviluppo.
- Sono definiti nei *profili*, che costituiscono uno dei principali meccanismi di estensione di UML.
- Alcuni profili disponibili:
  - CORBA
  - J2EE
  - SysML

# UML: diagrammi dei casi d'uso



# Contenuti

- Requisiti
  - Modello statico:
    - diagramma dei casi d'uso
  - Modello dinamico:
    - narrazioni associate ai casi d'uso

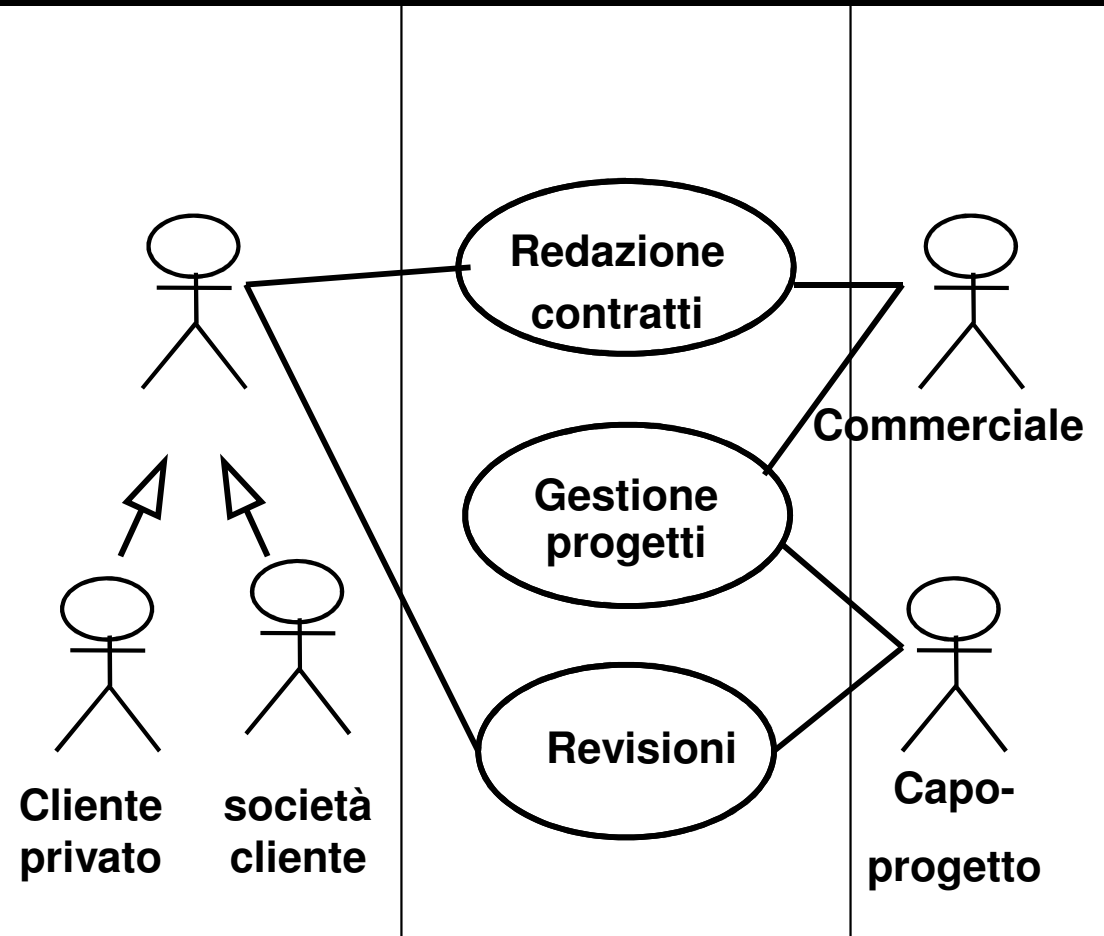
# Casi d'uso

- Il diagramma dei casi d'uso cattura il comportamento di un sistema, visto dall'esterno
- Un *attore* è un'entità esterna al sistema, che interagisce con esso in un determinato ruolo
  - utente
  - altro sistema
- Un caso d'uso è
  - una funzionalità o servizio offerto dal sistema a uno o più attori.
  - espressa come un'insieme di SCENARI
- Uno scenario è
  - Una sequenza di interazioni (scambi di messaggi) tra sistema e attori

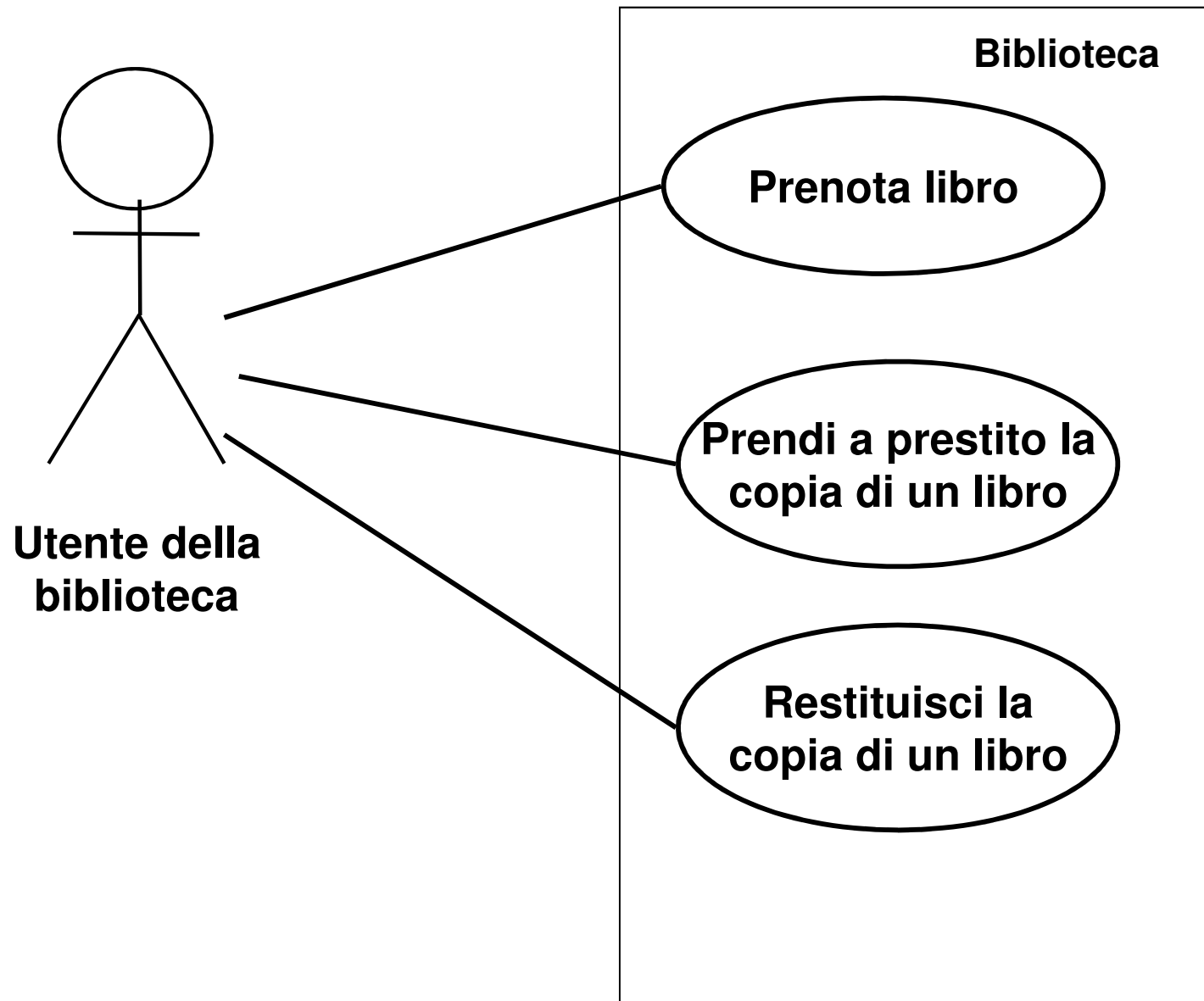
# Esempio di diagramma dei casi d'uso

## ■ Esempio

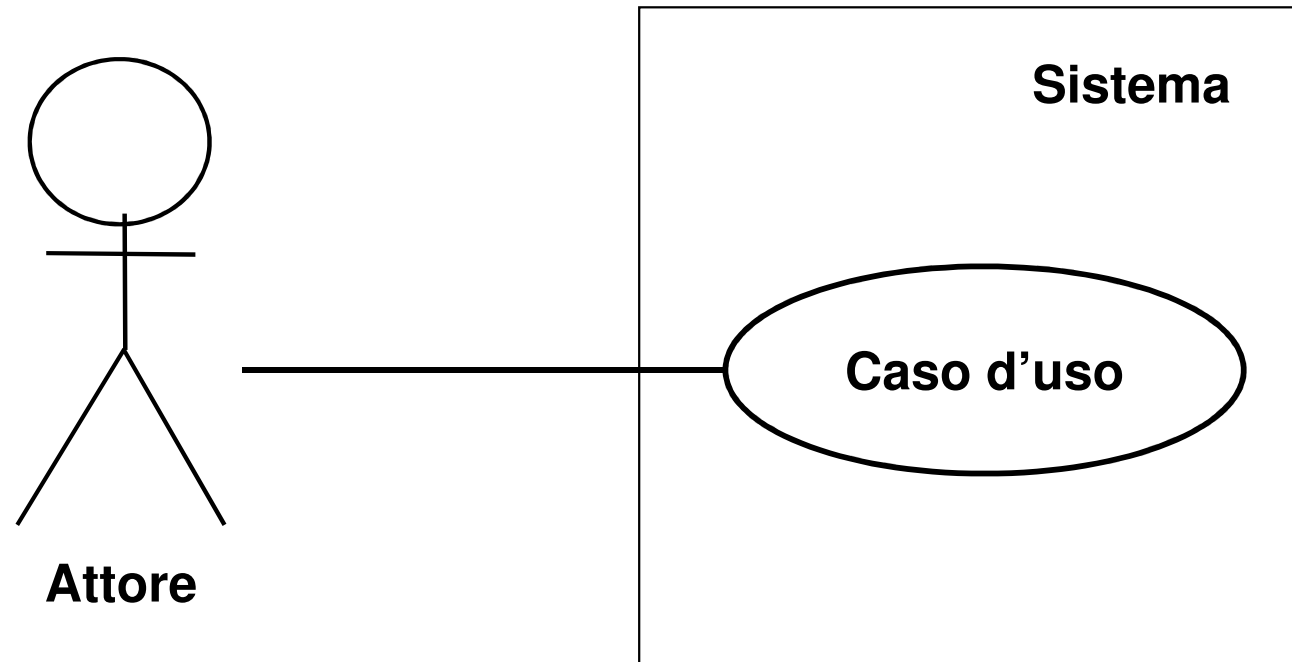
- I clienti possono essere privati o società
- Clienti e commerciali partecipano alla redazione dei contratti
- Clienti e capiprogetto partecipano alle revisioni
- Commerciali e capiprogetto gestiscono i progetti



# Caso d'uso: esempio

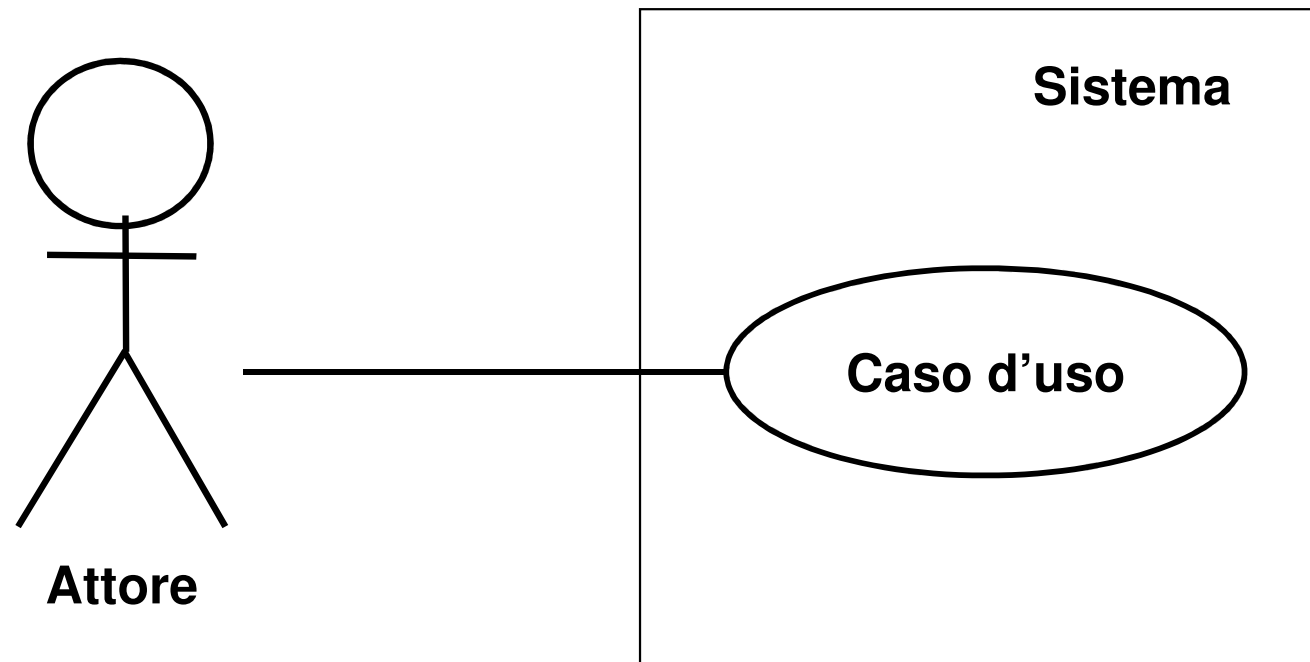


# Sintassi del caso d'uso



- Attore: omino con nome (UpperCamelCase) (è una classe)
- Associazione: senza nome
- Caso d'uso: ovale con nome (UpperCamelCase) (verbo)
- Sistema: rettangolo con o senza nome, contiene i casi d'uso
- L'associazione attori—casi d'uso è molti a molti.
- Casi d'uso senza attori solo se inclusi (included)

# Semantica del caso d'uso



- Un attore è un utente o 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)
  - Eventualmente Tempo

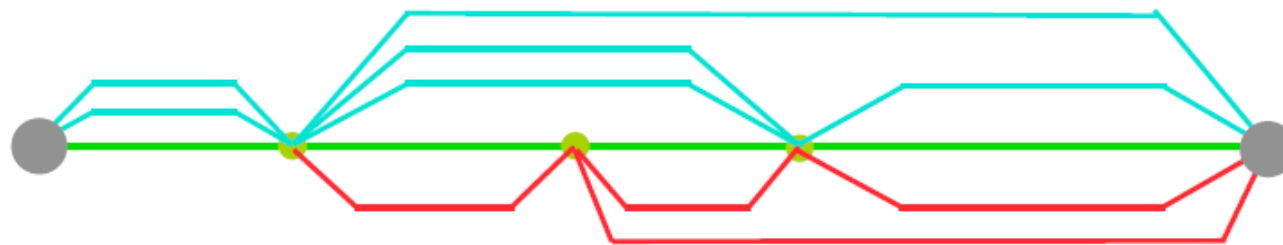
# Casi d'uso: descrizione narrativa

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

# Casi d'uso e scenari

– Ogni caso d'uso

- Ha una sequenza di transizioni normale o di base
- Può avere varie sequenze alternative
- Ha varie sequenze di transazioni eccezionali per la gestione di situazioni erranee



- Uno scenario è un'istanza di un caso d'uso, una storia, una sequenza di azioni che produce un risultato osservabile ad uno o più attori.



# Descrizione narrativa

*Nome:  
ID*

Nome del caso d'uso  
Numero

*Breve descrizione:*

Due righe

*Attori primari:*

Attori che avviano il caso d'uso

*Attori secondari:*

Altri attori che interagiscono con il caso d'uso

*Precondizioni:*

Devono valere prima dell'esecuzione del caso d'uso

*Sequenza degli  
eventi principale:*

Sequenza di passi

*Postcondizioni:*

Della sequenza principale

*Sequenze alternative  
degli eventi:*

Errori, ramificazioni e interruzioni nella sequenza principale

# Esempio di descrizione narrativa

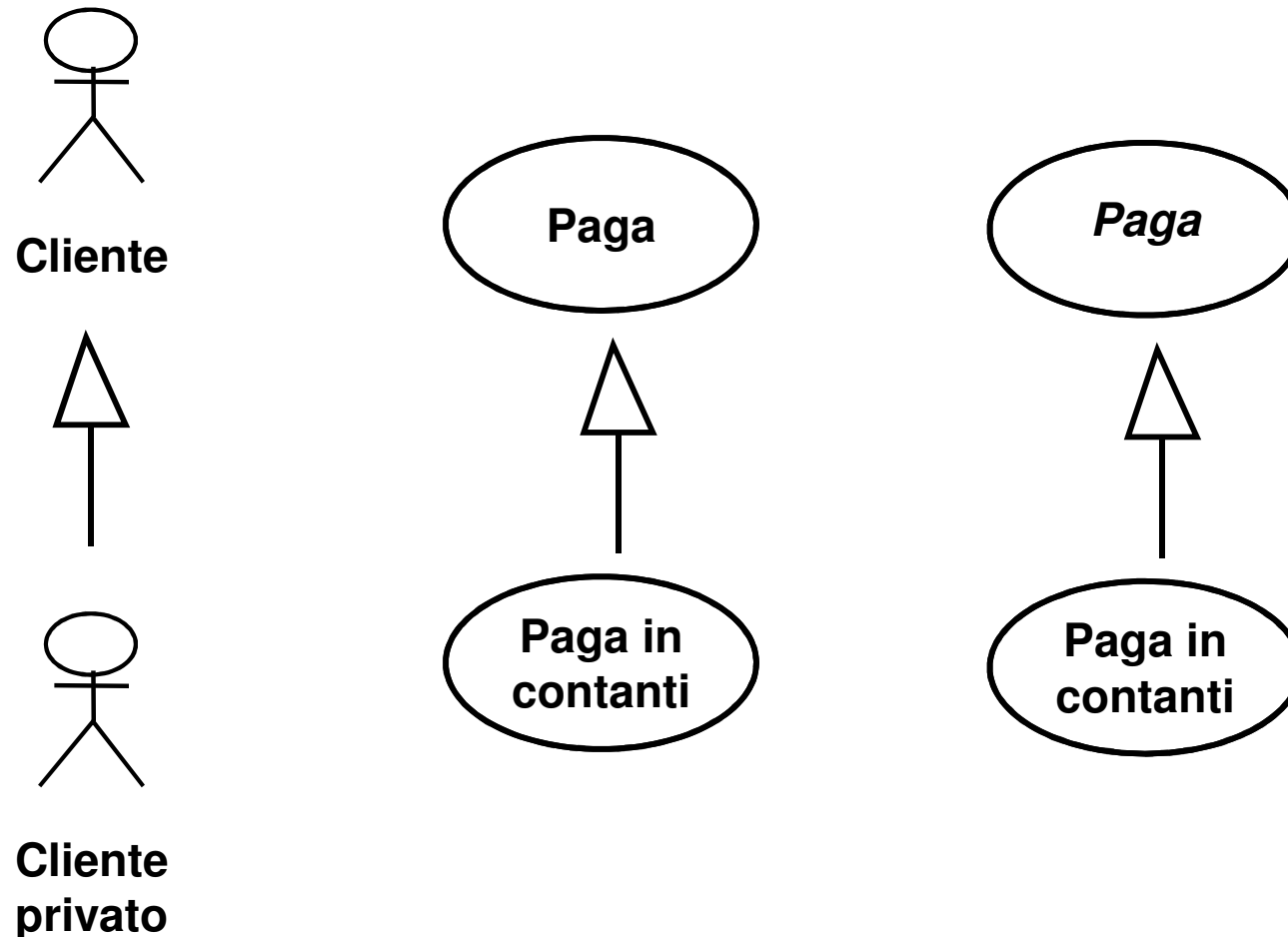
<b>Nome:</b>	Estendi il prestito
<b>Breve descrizione:</b>	Un utente desidera prolungare il periodo di prestito
<b>Attori primari:</b>	Utente della biblioteca
<b>Attori secondari:</b>	Nessuno
<b>Precondizioni:</b>	Il libro è già assegnato all'utente
<b>Sequenza degli eventi principale:</b>	<ol style="list-style-type: none"><li>1. Controllare che nessuno l'abbia prenotato</li><li>2. Controllare lo stato dell'utente</li><li>3. Aggiornare utente e stato libro</li></ol>
<b>Postcondizioni:</b>	Prestito esteso, stato utente aggiornato
<b>Sequenze alternative degli eventi:</b>	Il libro è già prenotato, utente inaffidabile

# *Sequenza degli eventi principale: se, per, while*

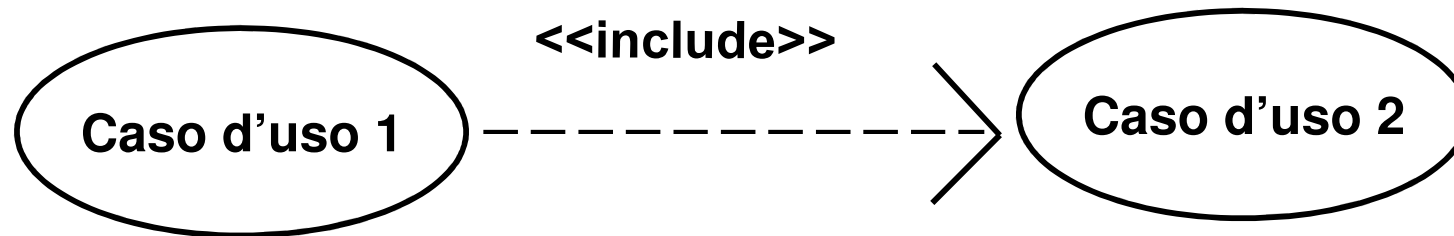
1. Fare a
2. Se (espressione booleana)
  1. Fare b
3. Altrimenti
  1. Fare c [OPZIONALE]
4. Per (espressione di iterazione)
  - 4.1 Fare d
5. While (espressione booleana)
  - 5.1 Fare e
  - 5.2 Fare f

# Casi d'uso: generalizzazione

è possibile il caso d'uso (o l'attore) padre sia astratto

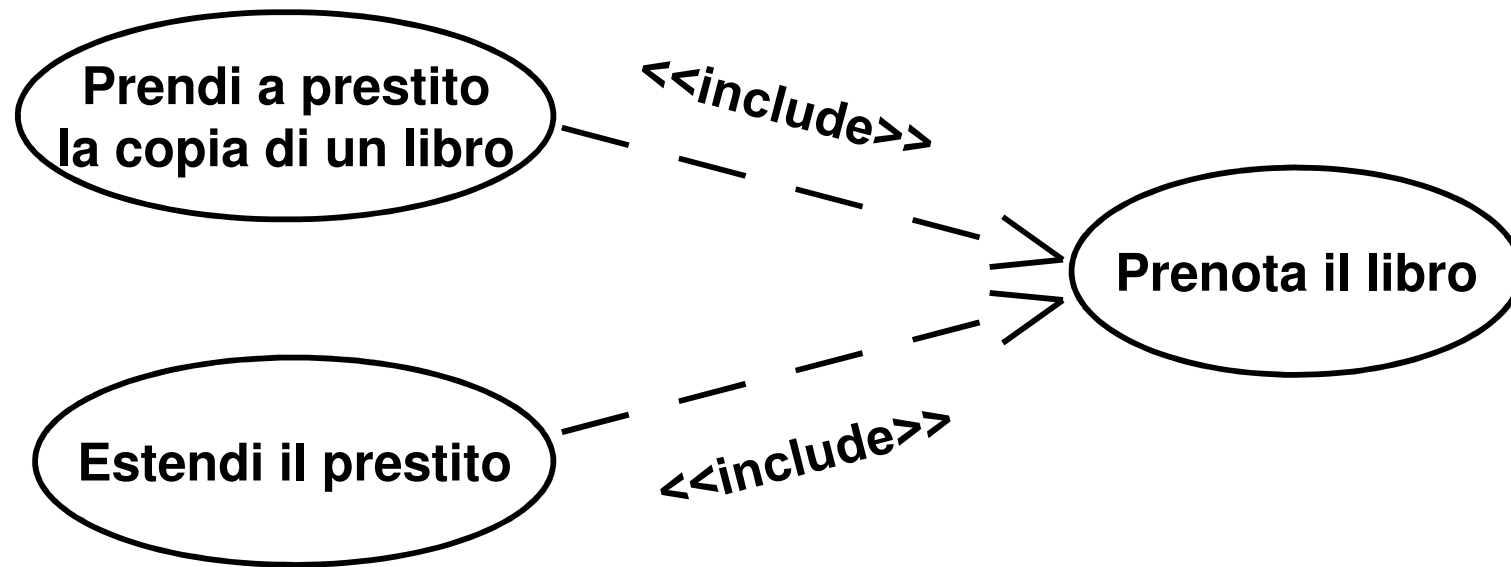


# Inclusione di caso d'uso



- 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)

# 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. Fare a
  2. Fare b
  3. Include (nome caso incluso)
  4. Fare d
- Il caso d'uso incluso è specificato a parte come caso base
    - Può essere istanziabile (completo) (avviato da un attore)
    - ....o non istanziabile (eseguito solo quando incluso)

# "Per casa": Chiavi magnetiche

- Documento dei requisiti; Problem diagram; Casi d'uso
- 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.



# Syllabus

Arlo w capp 4 e 5.