

Introduzione a UML

UML: Casi d'Uso e narrative

Roberta Gori 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

- Astrazioni del sistema (o del dominio) usati per specificarne struttura e comportamento
- Un modello cattura aspetti importanti della cosa da modellare da un certo punto di vista e semplifica o omette il resto:
esempio pianta di una casa
- Un modello e' espresso con un formalismo che rende facile usarlo e comprenderlo

Proprieta' di un modello

- Il modello fornisce una rappresentazione del sistema che e' ridotta al minimo per minimizzarne la complessita' di gestione
- Un sistema e' 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

- 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?

Come usare un modello

- Un progetto può essere realizzato come

- **Abbozzo (sketch)**

il modello non e' completo, possono rendere visibili soluzioni alternative

- **Progetto dettagliato (blueprint)**

contengono dettagli sufficienti per permettere agli sviluppatori di creare un ready-to-run system senza prendere decisioni di progettazione

Modelli come abbozzo o blueprint possono essere usati per forward or backward engineering

- **Eseguibile (UML come linguaggio di programmazione)**

talmente completo e preciso da permettere di generare il codice automaticamente dal modello

Come descrivere il modello

- Bisogno di un linguaggio comune
- Semi-formale
- Che possa essere utilizzato per descrivere aspetti diversi del progetto:
 - Descrizione strutturale vs descrizione di un comportamento
 - Bozza di progetto vs progetto finale
 - Descrizione del dominio vs descrizione della struttura del codice

UML

Unified Modeling Language: linguaggio di modellazione unificato

- Famiglia di notazioni grafiche basate su un singolo meta-modello
- Serve per supportare la descrizione e il progetto di sistemi software in particolare applicazioni sw OO ma non solo....
- I diagrammi sono generalmente facilmente comprensibili agli utenti con un minimo di conoscenze nel campo
- UP(Unified Process) 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
- 2006: UML 2.0
- 2006 . . . : Model Driven Architecture (MDA)
- Attualmente UML 2.5

Generalità

- Unificazione a livello di linguaggio, non di metodo
- 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

Obiettivi

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

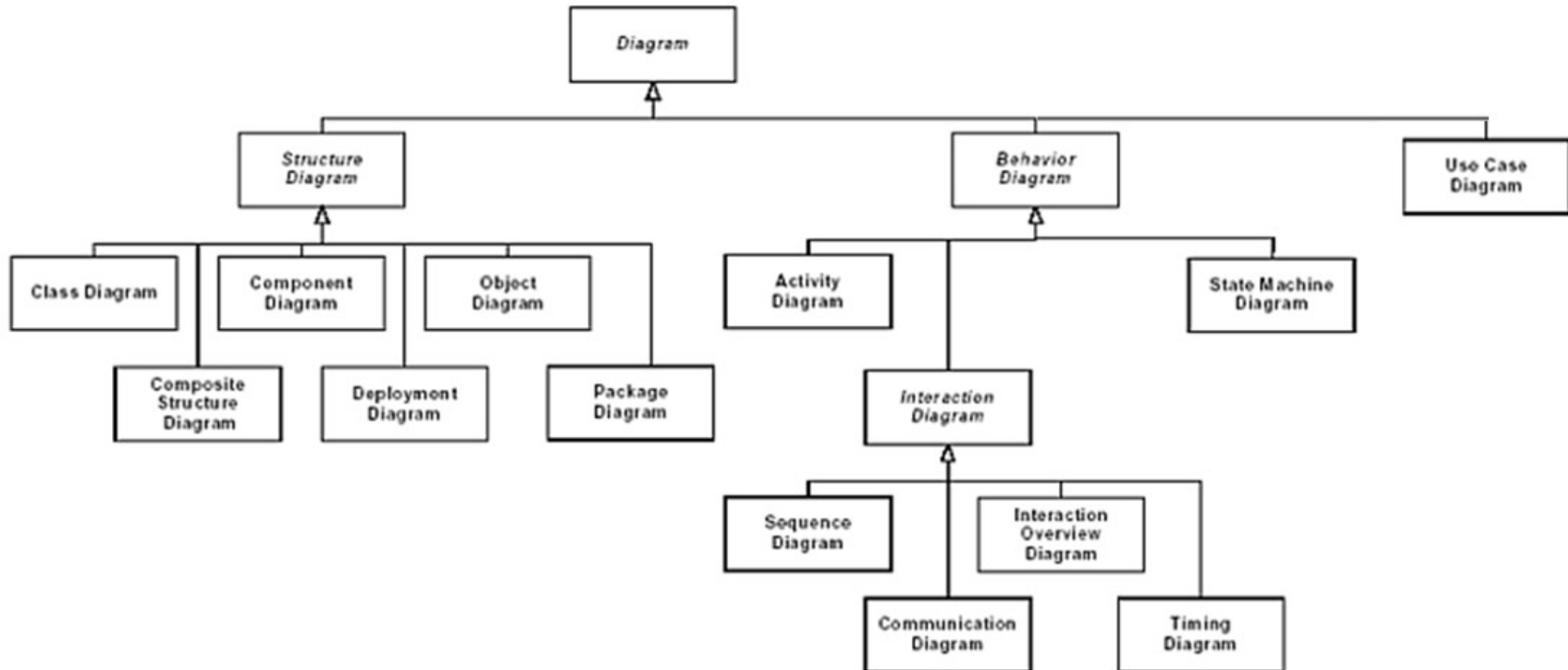
su UML

- 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

Diagrammi in UML

- In UML un modello è rappresentato graficamente come un diagramma.
- Un diagramma ci dà una vista della parte di realtà descritta da un modello.
- Modelliamo, per mezzo di opportuni **diagrammi** le funzionalità , i processi e le architetture software di un progetto.
- UML offre una molteplicità di diagrammi: **molteplicità di viste** per ogni modello (per esempio dominio e struttura del codice) che interessa descrivere di una applicazione.
- Un **insieme di diagrammi** descrive un modello.

Diagrammi in UML: 14 diversi



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 che vedremo nel corso

- ...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

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
- Vista: descrizione di un aspetto di un modello

Elementi base di UML

■ Diagrammi

- delle classi, dei casi d'uso, d'interazione, ...

■ Entità

- classi, interfacce, componenti, casi d'uso, ...

■ Relazioni

- associazioni, generalizzazioni, dipendenze, ...

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

UML: diagrammi dei casi d'uso

Contenuti

- Modellazione dei Requisiti: consiste nel raccogliere e modellare i 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 descrive i requisiti funzionali del sistema
- Questo diagramma 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
- Un caso d'uso è
 - una funzionalità o 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

Modellazione dei casi d'uso

La modellazione prevede i seguenti passi:

- Individuare un potenziale confine del sistema
- Individuare gli attori
- Individuare i casi d'uso
- Individuare le relazioni attore-caso d'uso

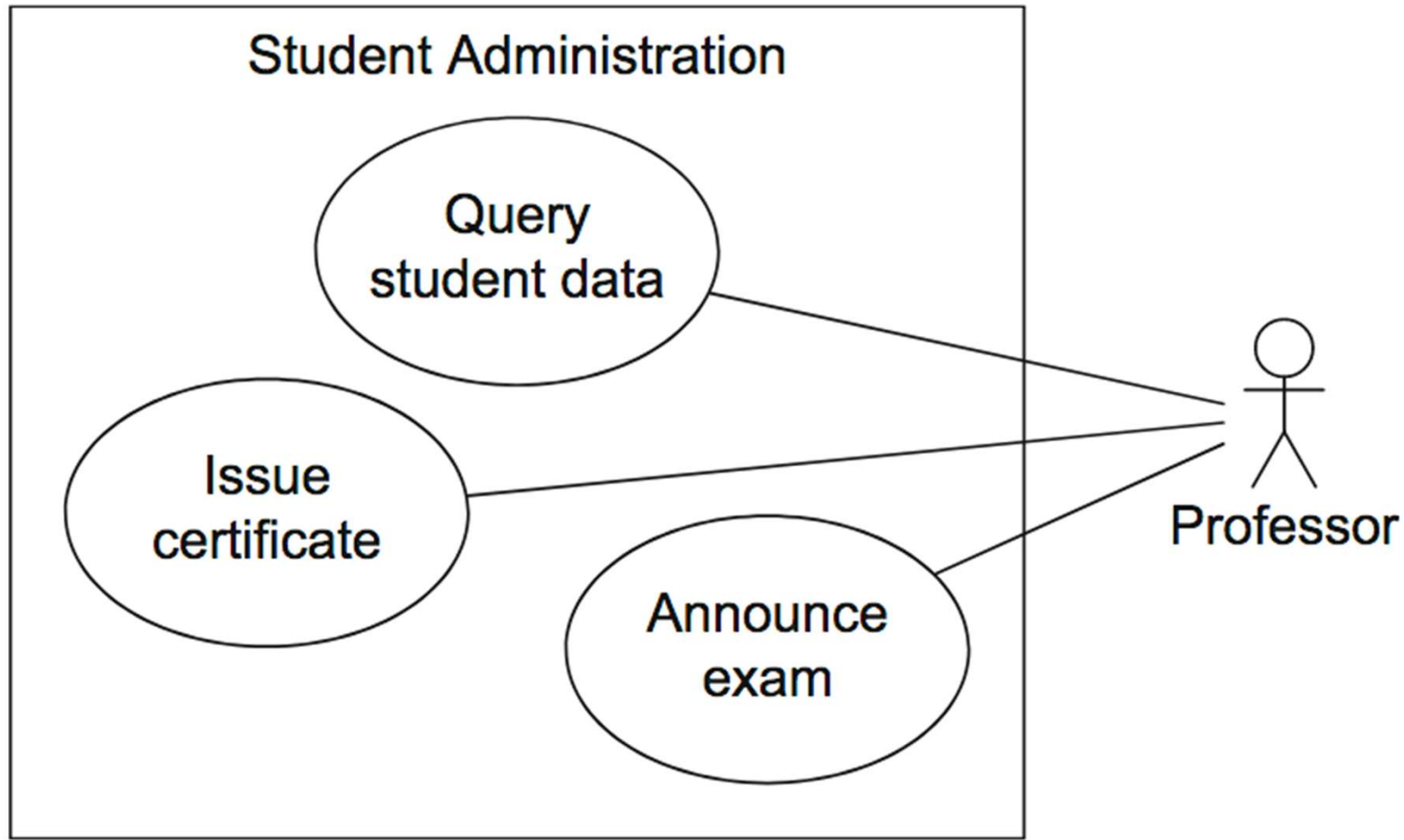
- Specificare il caso d'uso
 - con una descrizione testuale (narrativa)

Modellazione dei casi d'uso

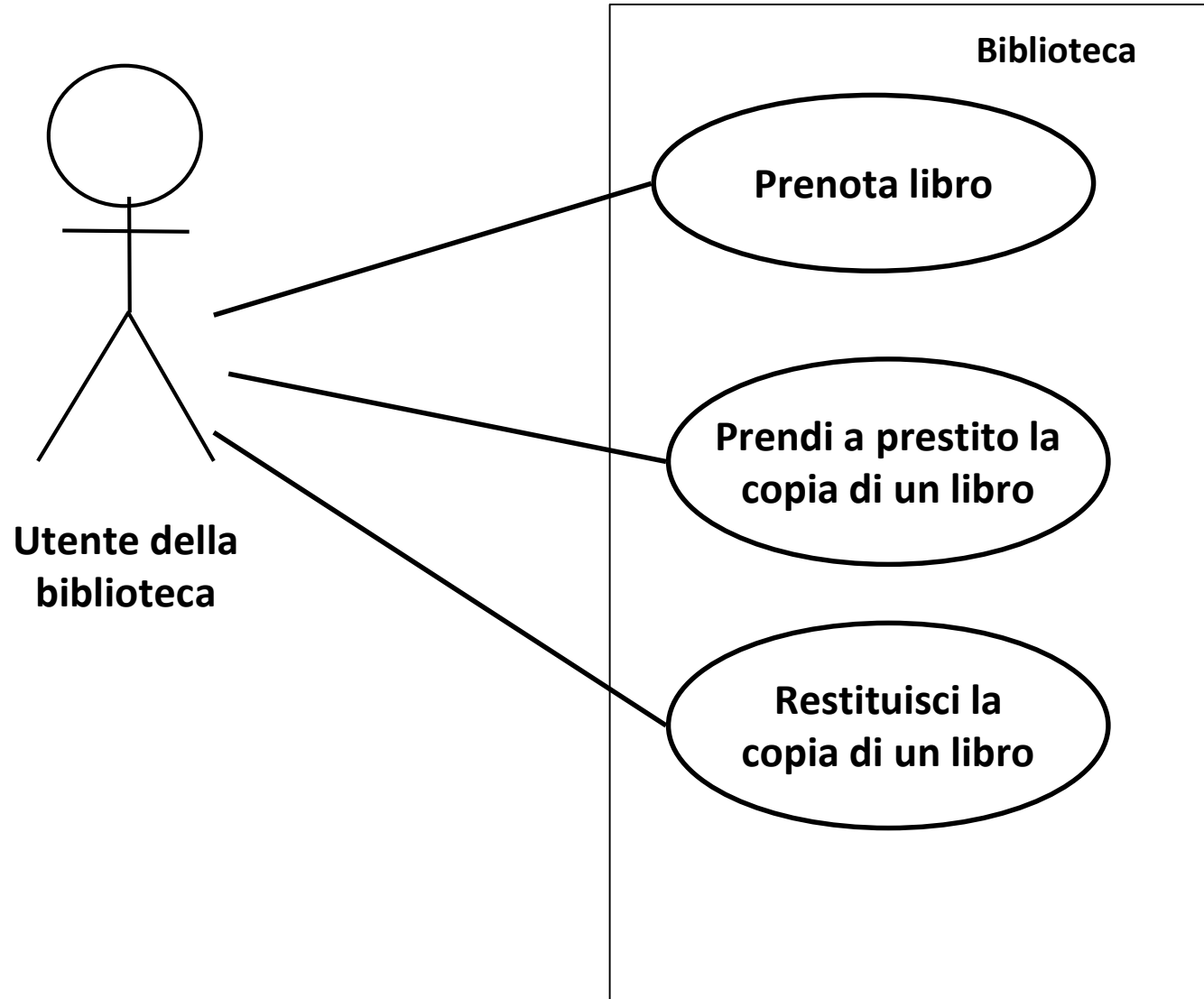
Il diagramma è composto da quattro entità:

- **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



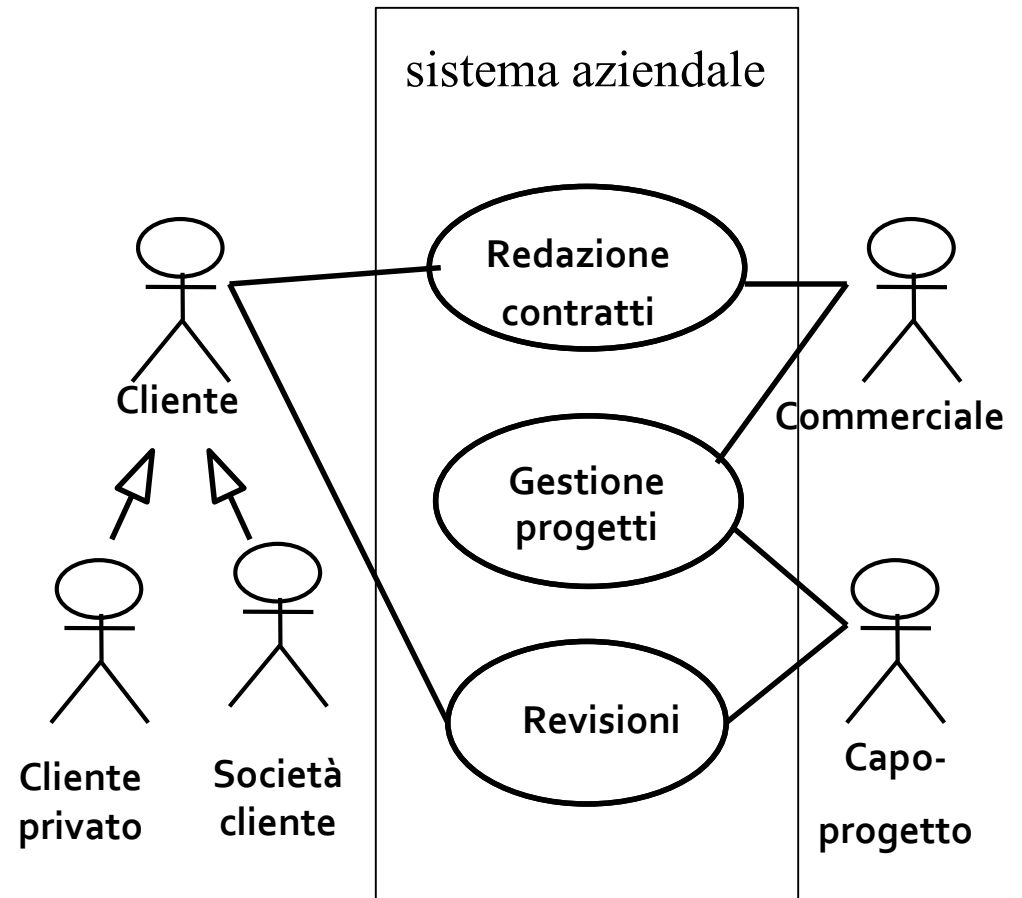
Altro esempio di diagramma dei casi d'uso



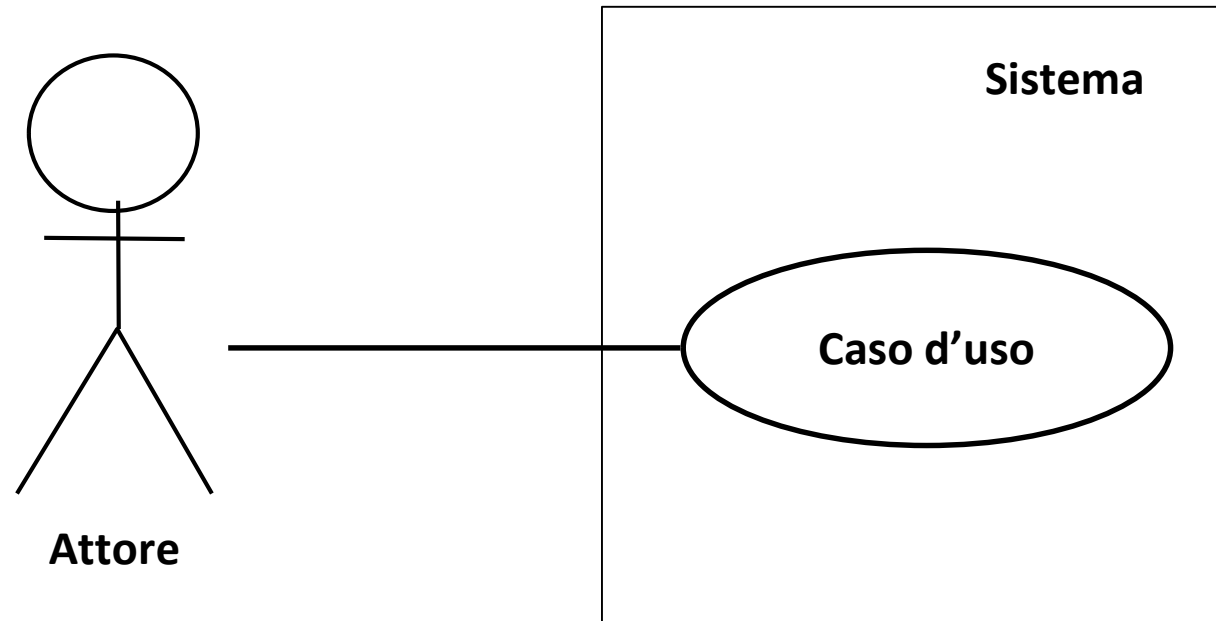
Terzo esempio di diagramma dei casi d'uso

■ Esempio

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

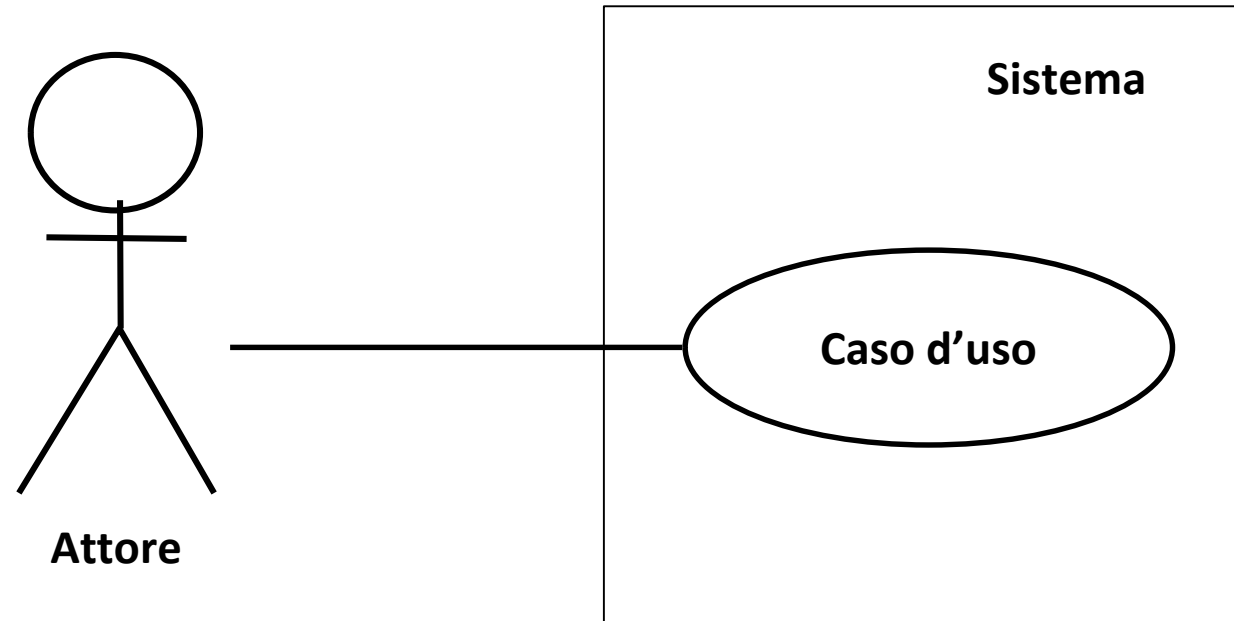


Sintassi del caso d'uso



- 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



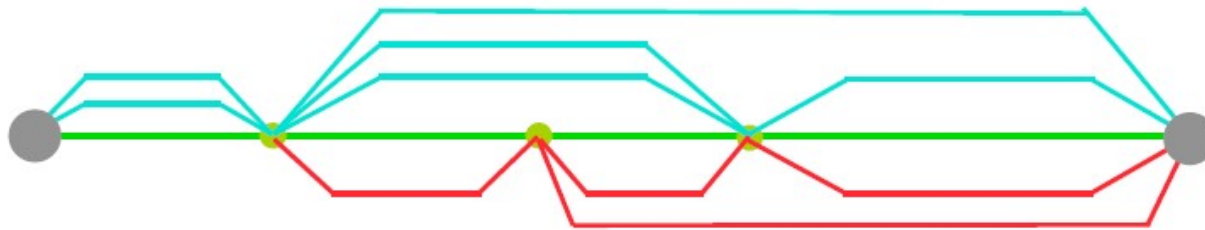
- Un attore è un utente o un altro sistema, in un particolare ruolo
- Un caso d'uso è un compito (task) che gli attori eseguono con l'ausilio del sistema
- L'associazione attori—casi rappresenta un'interazione (sequenza di messaggi).
- Un caso d'uso è iniziato SOLO da un attore (principale: la narrativa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

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 identificatore
Breve descrizione:	Due righe riassuntive
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

Nome:	Estendi il prestito
Breve descrizione:	Un utente desidera prolungare il periodo di prestito
Attori primari:	Utente della biblioteca
Attori secondari:	Nessuno
Precondizioni:	Il libro è già assegnato all'utente
Sequenza degli eventi principale:	<ol style="list-style-type: none">1. Accerta che nessuno l'abbia prenotato2. Accerta che l'utente sia ammesso al prestito3. Aggiorna utente e stato del libro
Postcondizioni:	Prestito esteso, stato utente aggiornato
Sequenze alternative degli eventi:	Il libro è già prenotato, utente non ammesso al prestito

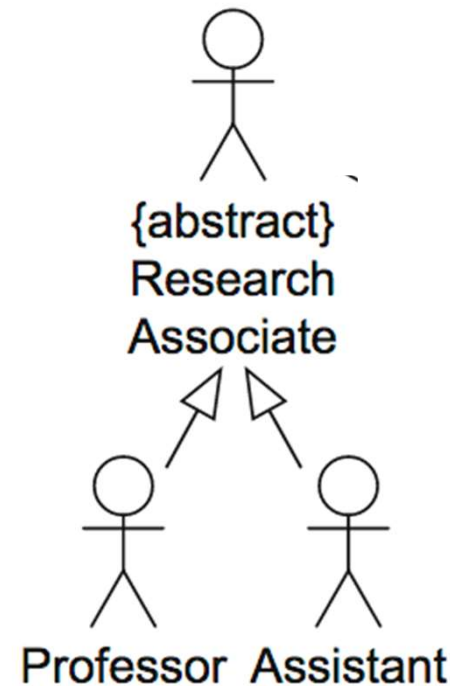
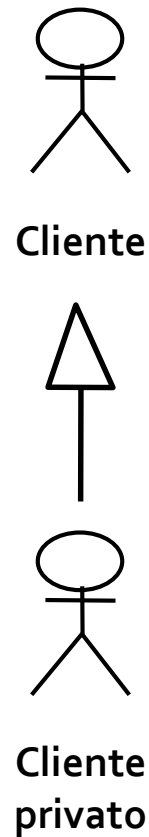
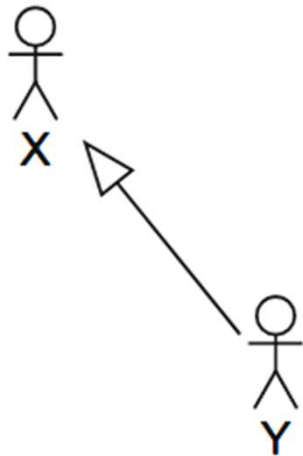
Relazione tra preconditione, postcondizione, sequenza principale e sequenza alternativa

- Parallelo con le triple di Hoare:
- {Precondizione} sequenza principale {Postcondizione}
- Quindi:
 - per ogni stato σ che soddisfa la preconditione, l'esecuzione del caso d'uso a partire da σ termina producendo uno stato σ' 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

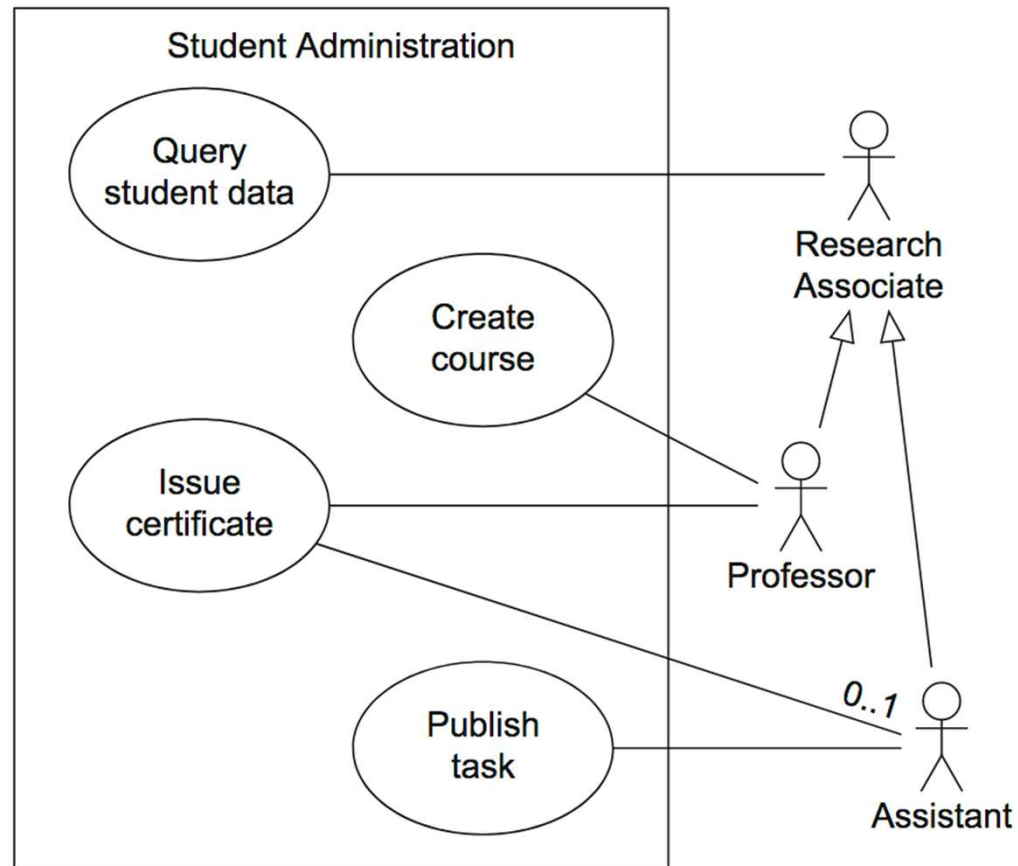
1. Fare a
2. Se (espressione booleana)
 1. Fare b
3. Altrimenti
 1. Fare c [OPZIONALE]
4. Per (espressione di iterazione)
 - 4.1 Fare d
5. While (espressione booleana)
 - 5.1 Fare e
 - 5.2 Fare f

Casi d'uso: generalizzazione tra attori

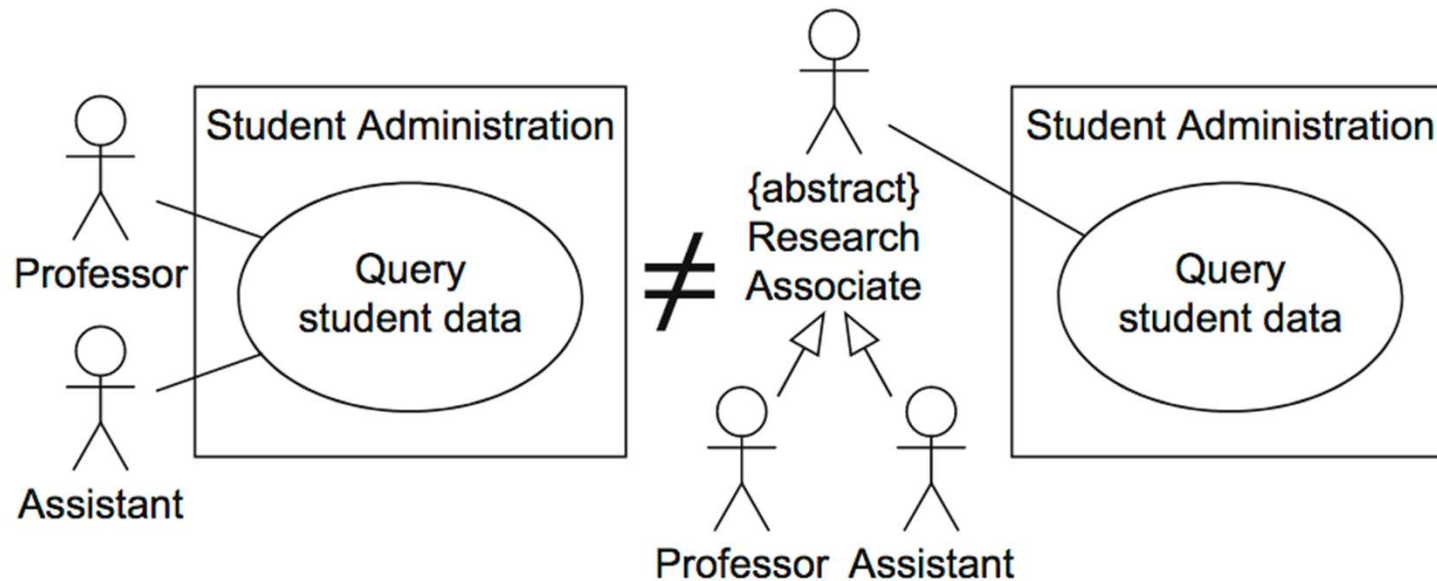


È possibile che l'attore padre sia astratto

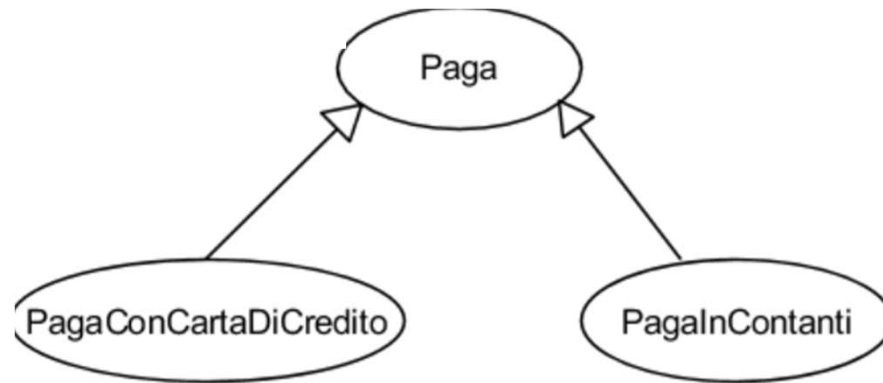
Esempi di generalizzazione



Esempi di generalizzazione

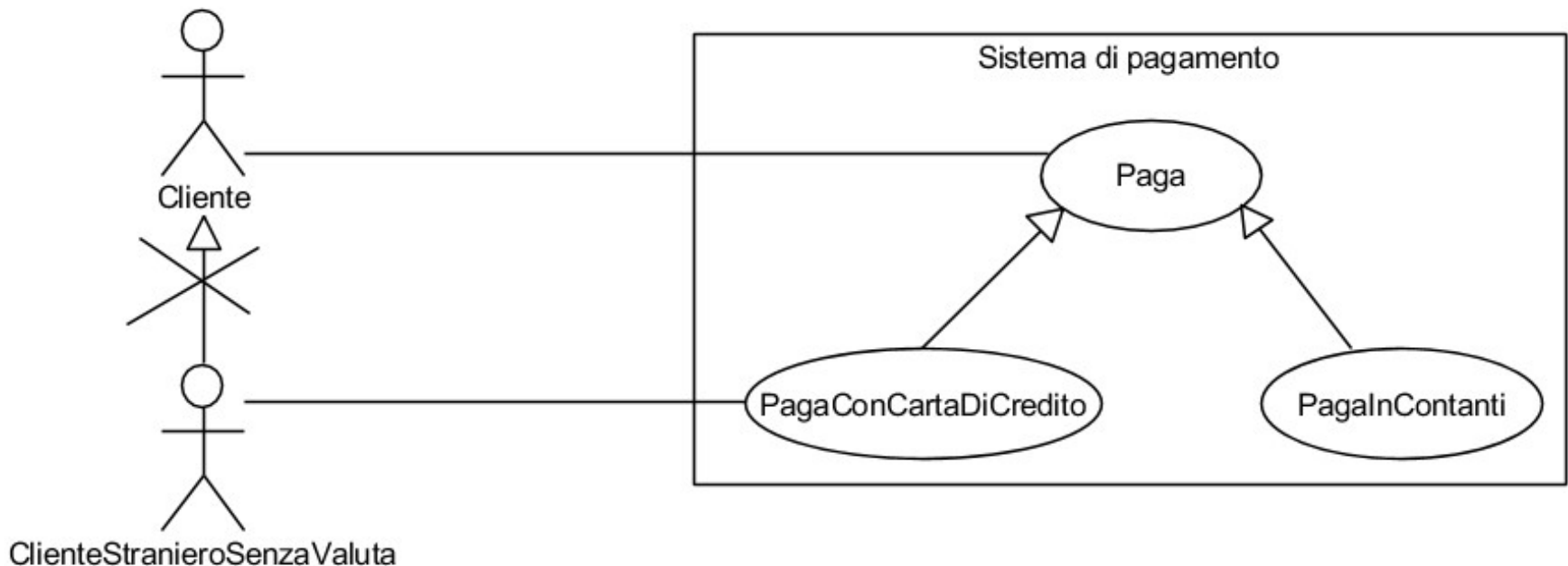


Generalizzazione dei casi d'uso

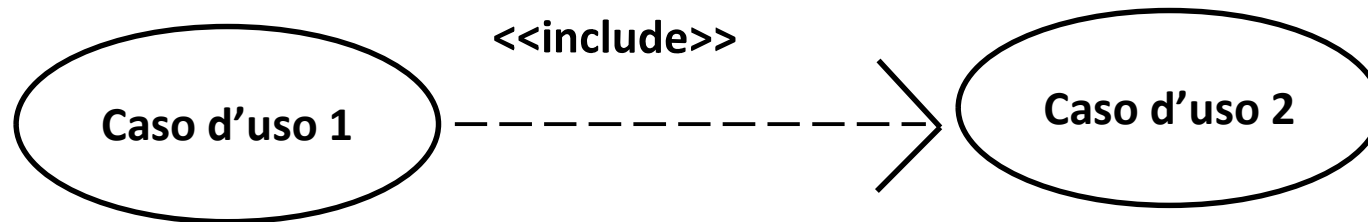


Attenti alla generalizzazione...

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

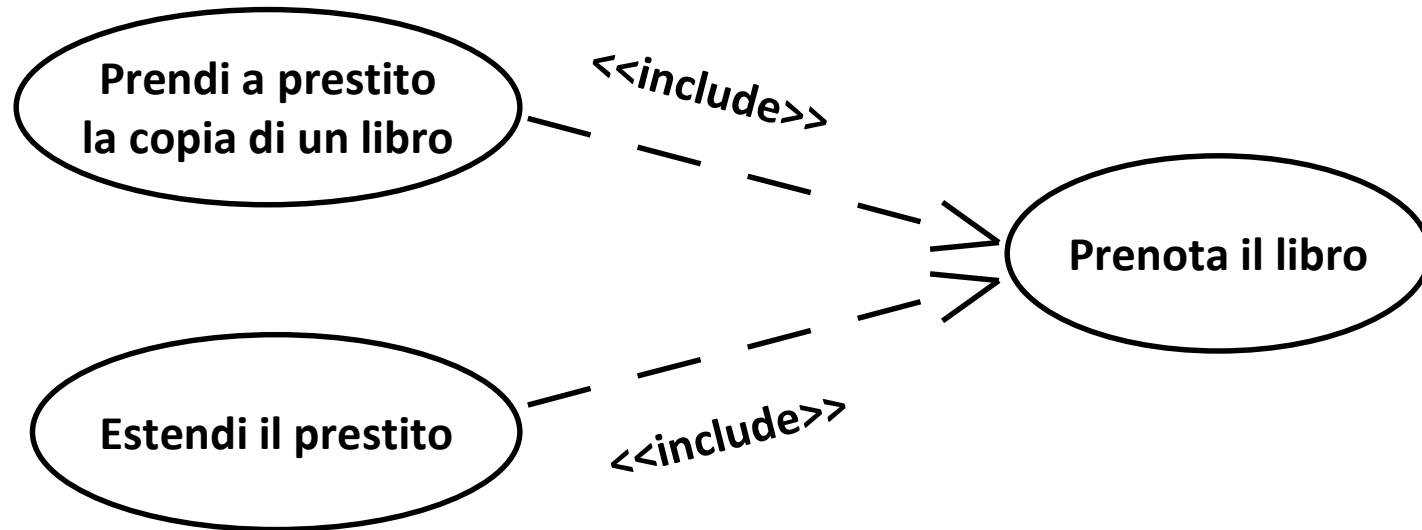


Inclusione di caso d'uso



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

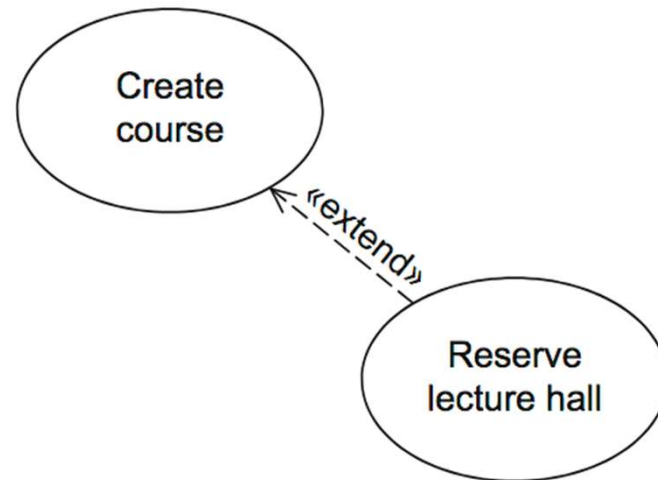
Estensione di un caso d'uso



- 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 puo'** riservare un aula ma **non e' obbligatorio**

Syllabus

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

Cap 1-2-3

"Per casa": Chiavi magnetiche

- 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.