

3. Modellazione con UML

Paola Barra
a.a. 2022/2023

Gestire la complessità

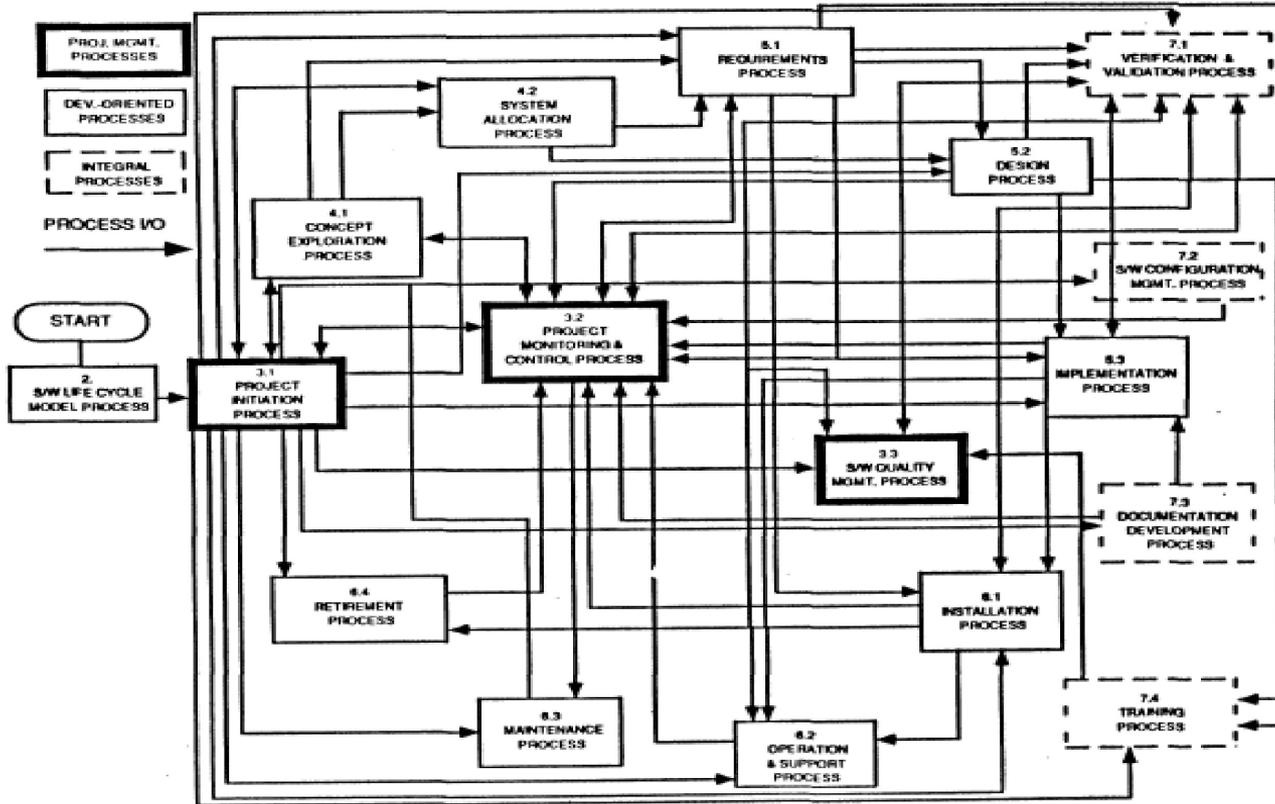
Astrazione e modellazione

Decomposizione

Gerarchia

Astrazione e modellazione

Qual è il problema di questo disegno?



Astrazione

- I sistemi complessi sono difficili da capire
 - Fenomeno 7 +- 2
 - La nostra memoria a breve termine non può memorizzare più di 7 +- 2 pezzi allo stesso tempo -> limite del cervello
 - Telefono Uniparthenope: 390815446543
 - Spezzettamento
 - Collezione di gruppi di oggetti per ridurre la complessità
 - 4 pezzi:
 - Codice paese, codice città, codice Uniparthenope, Ufficio



Astrazione

- L'astrazione consente di ignorare dettagli inutili

Cos'è l'astrazione?

- *Un processo del pensiero dove le idee sono separate dagli oggetti*

Astrazione come attività

- *Idea risultante di un processo di pensiero dove un'idea è stata separata da un oggetto*

Astrazione come entità

- Le idee possono essere espresse dai **modelli**

Decomposizione

Gestire la complessità

- Una tecnica usata per padroneggiare la complessità (*'divide and conquer'*)
- Due tipi di decomposizione
 - Decomposizione funzionale
 - Decomposizione orientata agli oggetti

Decomposizione

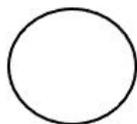
- **Decomposizione funzionale**
 - Il sistema è decomposto in moduli
 - Ogni modulo è una funzione principale nel dominio applicativo
 - I moduli possono essere decomposti in moduli più piccoli
- **Decomposizione orientata agli oggetti**
 - Il sistema è decomposto in classi ('oggetti')
 - Ogni classe è un'entità principale nel dominio applicativo
 - Le classi possono essere decomposte in classi più piccole

Decomposizione orientata agli oggetti vs funzionale

Qual è quella giusta?

Decomposizione funzionale

- La funzionalità è distribuita su tutto il sistema
- Il manutentore deve capire l'intero sistema per poter fare anche una singola modifica
- Conseguenza
 - Il codice sorgente è difficile da capire
 - Il codice sorgente è complesso e impossibile da mantenere
 - L'interfaccia utente spesso è scomoda e non intuitiva
- Esempio: *Forme* di Powerpoint
 - Come possiamo cambiare un quadrato in un cerchio?



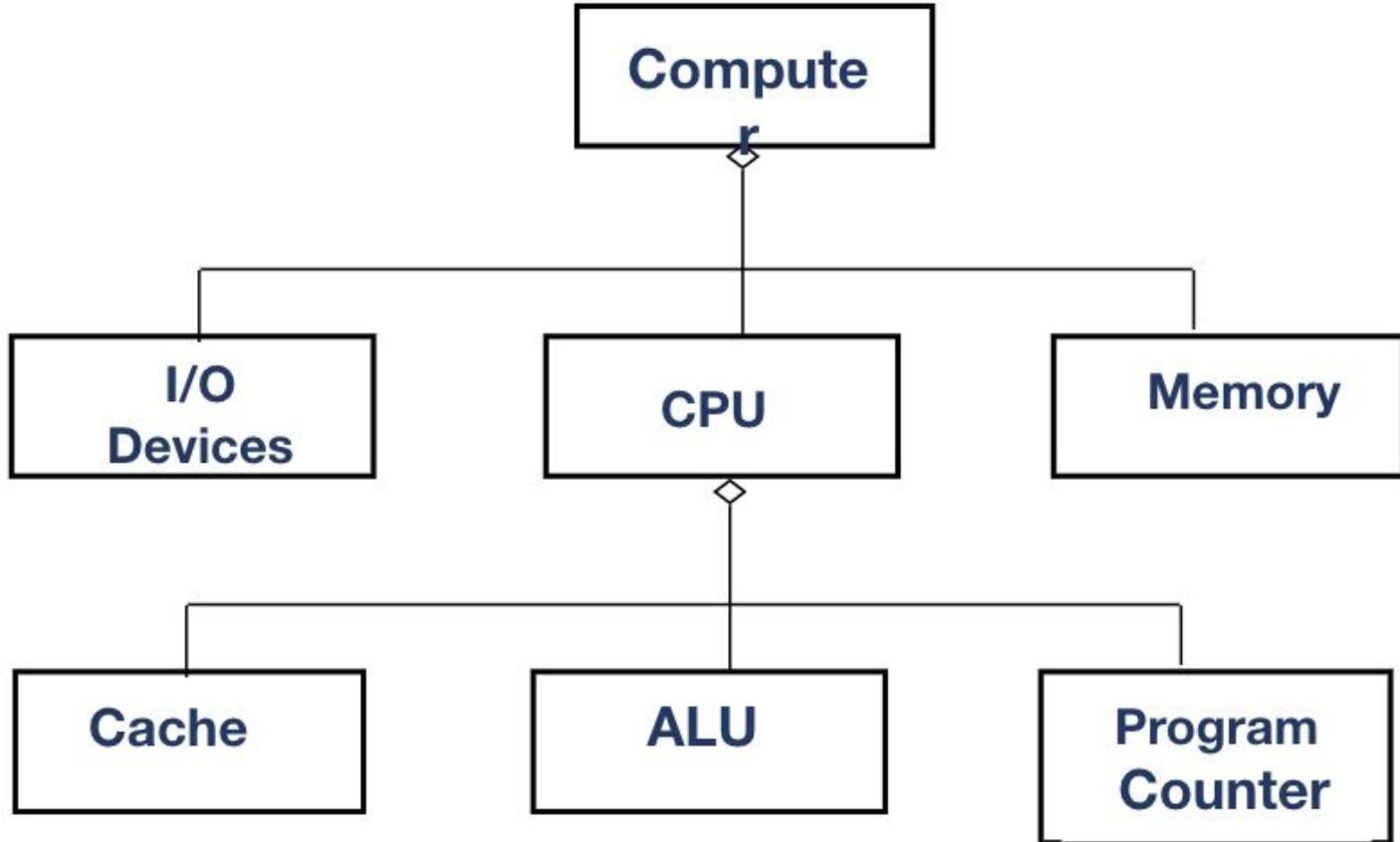
Gerarchia

Gerarchia

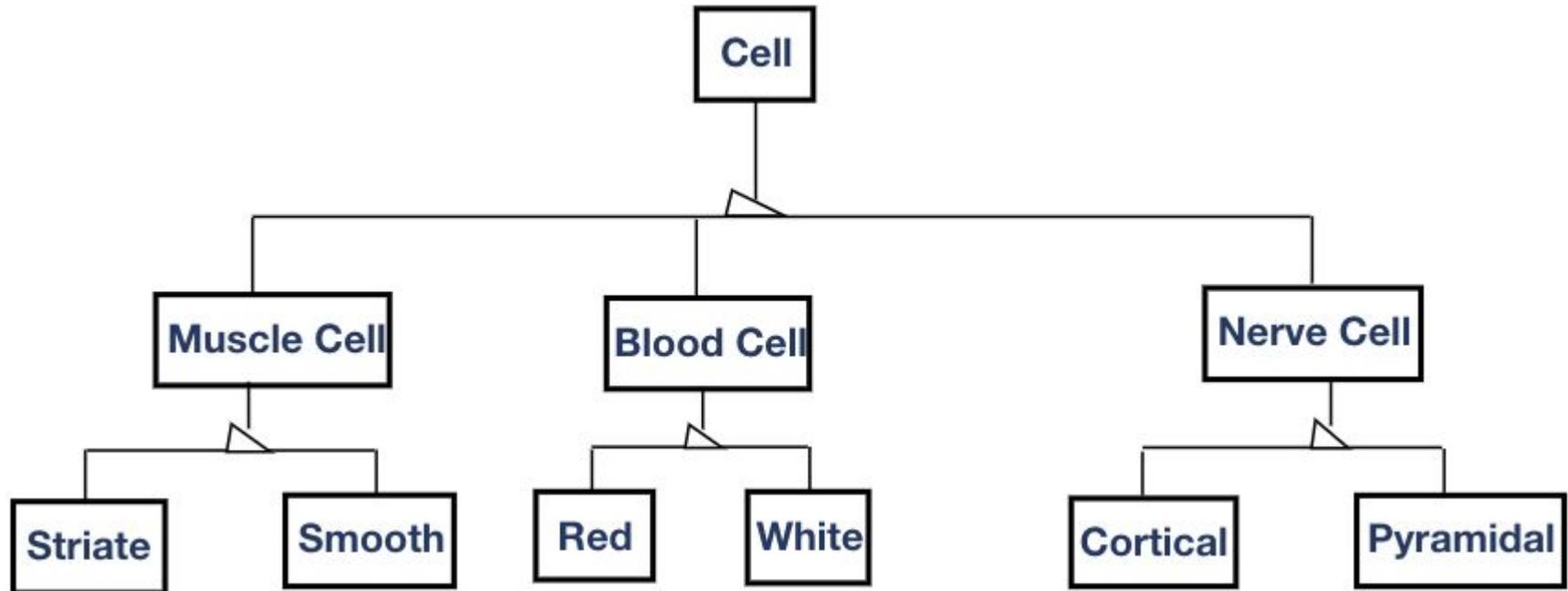
Una delle relazioni più importanti sono le gerarchie

- Due gerarchie speciali
 - La gerarchia '*parte di*'
 - La gerarchia '*è tipo di*'

Gerarchia 'parte di'



Gerarchia 'è tipo di'



Introduzione alla notazione UML

Diagrammi dei casi d'uso

Diagrammi delle classi

Diagrammi delle sequenze

Diagrammi degli stati

Diagrammi delle attività

A che punto siamo?

- Tre modi per trattare la complessità
 - Astrazione, decomposizione, gerarchia
- La decomposizione orientata agli oggetti è utile
 - Sfortunatamente, a seconda dello scopo del sistema, possono essere trovati oggetti diversi
- Come possiamo procedere nel modo giusto?
 - Iniziamo con una descrizione delle funzionalità di un sistema
 - Descriviamo poi la sua struttura

Organizzare le attività di sviluppo

- Ciclo di vita del software

Concetti e fenomeni

- **Fenomeno**

- Un oggetto nel mondo di un dominio così come lo percepiamo
 - Esempi: questa lezione alle 10:05, il mio orologio nero

- **Concetto**

- Descrive le proprietà comuni di un fenomeno
 - Esempio: tutte lezioni di ingegneria del software
 - Esempio: tutti gli orologi neri

- **Un concetto è una 3-tupla:**

- Nome: il nome distingue il concetto da altri concetti
- Scopo: proprietà che determinano se un fenomeno è membro di un concetto
- Membri: l'insieme di fenomeni che sono parte del concetto

Concetti, fenomeni, astrazioni e modellazione

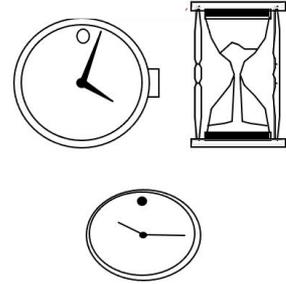
Nome

Orologio

Scopo

Un dispositivo che misura il tempo

Membri



- **Astrazione** definizione

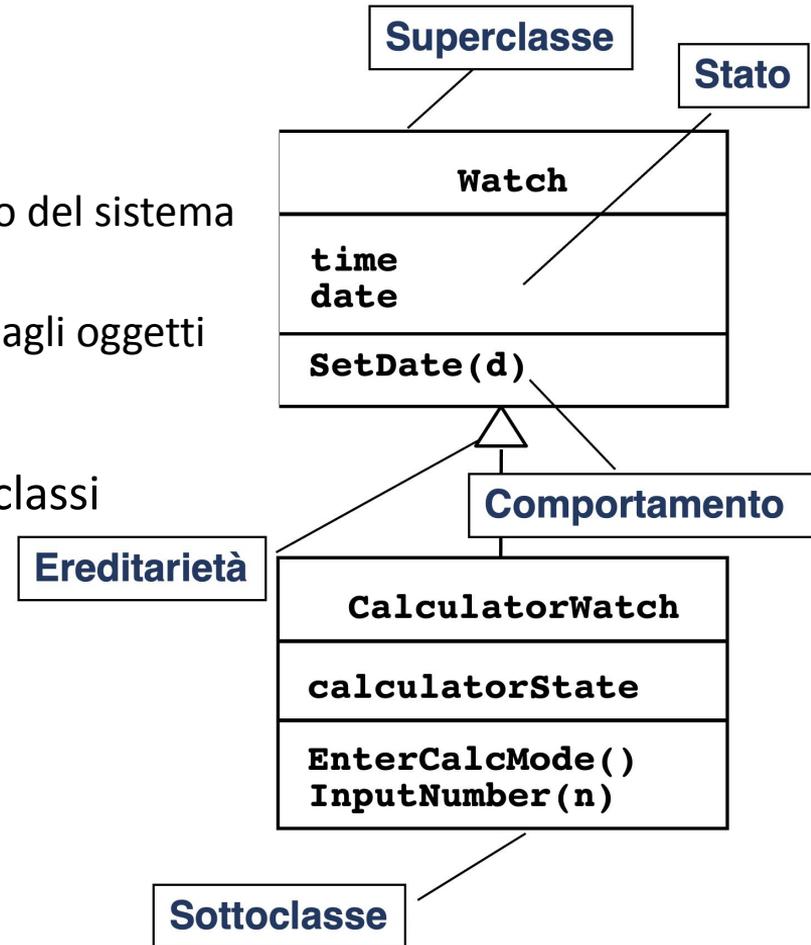
- Classificazione di fenomeni

- **Modellazione** definizione

- Sviluppo di astrazioni per rispondere a domande specifiche su un insieme di fenomeni ignorando dettagli irrilevanti

Tipi di dati astratti e classi

- Tipo di dato astratto
 - Un tipo la cui implementazione è nascosta al resto del sistema
- Classe
 - Un'astrazione nel contesto dei linguaggi orientati agli oggetti
 - Una classe incapsula stato e comportamento
 - Esempio: Orologio
- Differentemente dai tipi di dati astratti, le sottoclassi possono essere definite in termini di altre classi usando l'ereditarietà
 - Esempio: CalculatorWatch



Sistemi

- Un sistema è un insieme organizzato di parti comunicanti
 - Sistema naturale: un sistema il cui scopo ultimo non è noto
 - Sistema ingegnerizzato: Un sistema che è progettato e costruito da ingegneri per uno scopo specifico
- Le parti del sistema possono essere considerate a loro volta sistemi
 - Chiamate sottosistemi
- Esempi di sistemi naturali:
 - Universo, pianeta terra, oceano
- Esempi di sistemi ingegnerizzati:
 - Aeroplano, orologio, GPS
- Esempi di sottosistemi:
 - Motore aereo, batteria, satellite

Sistemi, modelli e view

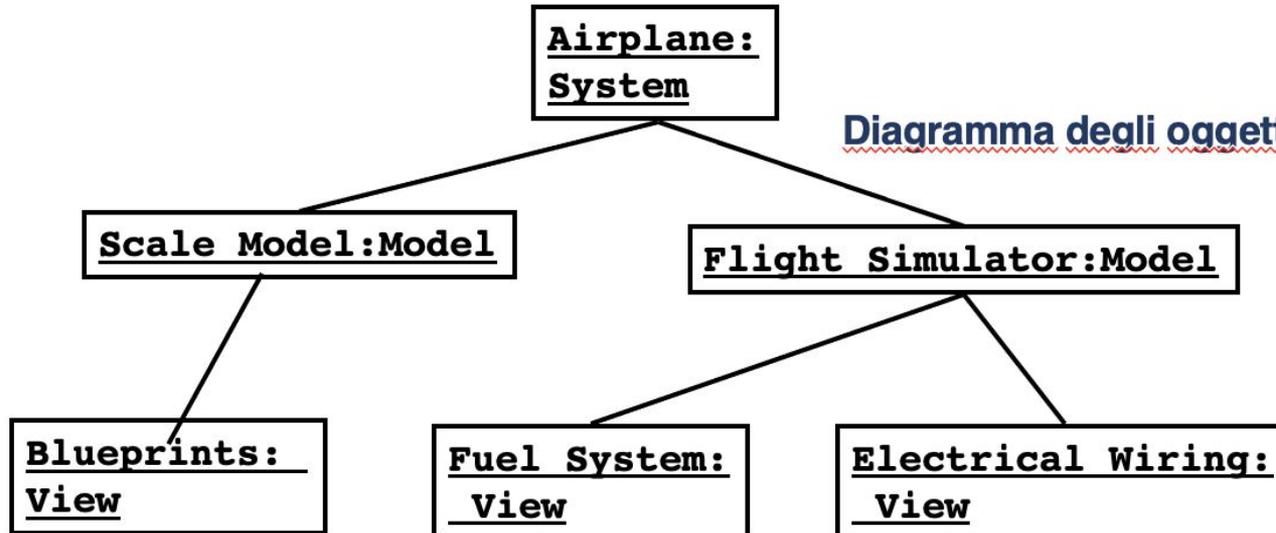
- Un modello è un'astrazione che descrive un sistema o sottosistema
- Una view raffigura aspetti selezionati di un modello
- Una notazione è un insieme di regole grafiche o testuali per illustrare modelli e view
 - Notazioni formali
- Sistema: aeroplano
- Modelli:
 - simulatore di volo
 - Modello in scala
- View:
 - Planimetria dei componenti di un aereo
 - Diagramma dei cavi elettrici, sistema del carburante, onda sonora creata dall'aeroplano

Sistemi modelli e view (UML)

Diagramma delle classi



Diagramma degli oggetti



Dominio dell'applicazione VS dominio della soluzione

- Dominio dell'applicazione (Analisi)
 - L'ambiente in cui il sistema sta funzionando
- Dominio della soluzione (Progettazione, Implementazione)
 - Le tecnologie usate per costruire il sistema
- Entrambi i domini contengono astrazioni che possiamo usare per la costruzione del modello del sistema

Cosa è l'UML?

UML (Unified Modeling Language)

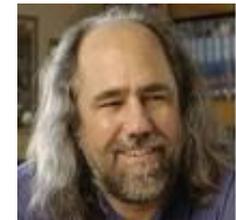
- Standard non proprietario per modellare sistemi software
- Convergenza delle notazioni usate nei metodi orientati agli oggetti
 - OMT (James Rumbaugh e colleghi)
 - Booch (Grady Booch)
 - OOSE (Ivar Jacobson)
- Versione corrente 2.5
 - Info sul portale OMG <http://www.uml.org/>
- Tool commerciali: rational (IBM), Together (Borland), Visual Architect (Business processes, BCD)
- Tool open source: ArgoUML, StarUML, Umbrello
- Commerciale e Open source: PoseidonUML (Gentleware)



25 year at GE Research, where he developed OMT, joined (IBM) Rational in 1994, CASE tool OMTool



At Ericsson until 1994, developed use cases and the CASE tool Objectory, at IBM Rational since 1995



Developed the Booch method (“clouds”), ACM Fellow 1995, and IBM Fellow 2003

Diagrammi UML

- **Diagrammi dei casi d'uso**
 - Descrivono il comportamento funzionale del sistema come sono visti dagli utenti
 - **Diagrammi delle classi**
 - Descrivono la struttura statica del sistema: oggetti, attributi, associazioni
 - **Diagrammi delle sequenze**
 - Descrivono il comportamento dinamico tra gli oggetti del sistema
 - **Diagrammi degli stati**
 - Descrivono il comportamento dinamico di un singolo oggetto
 - **Diagrammi delle attività**
 - Descrivono il comportamento dinamico di un sistema, in particolare il flusso di lavoro
-

UML: convenzione di base

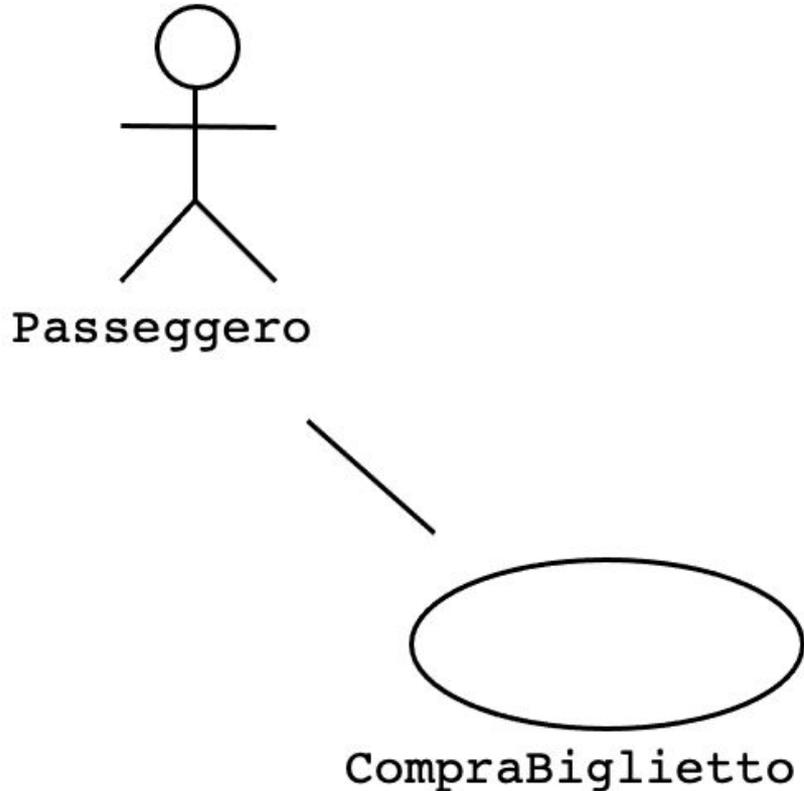
- Tutti i diagrammi UML denotano grafi di nodi e vertici
 - I nodi sono entità disegnati come rettangoli o ovali
 - I rettangoli denotano classi o istanze
 - Gli ovali denotano funzioni
- I nomi delle classi non sono sottolineati
 - SimpleWatch
 - Firefighter
- I nomi delle istanze sono sottolineati
 - myWatch:SimpleWatch
 - Joe:Firefighter
- Un arco tra due nodi indica una relazione tra le entità corrispondenti

Riepilogo della notazione UML base

- UML fornisce un'ampia varietà di notazioni per modellare molti aspetti dei sistemi software
- Ci siamo concentrati su poche notazioni
 - Modello funzionale: diagrammi dei casi d'uso
 - Modello ad oggetti: diagramma delle classi
 - Modello dinamico: diagrammi delle sequenze, degli stati

Diagramma dei casi d'uso

Diagrammi dei casi d'uso



- Usati durante la fase di scoperta dei requisiti per rappresentare il comportamento esterno
- Gli **attori** rappresentano ruoli, cioè, un tipo di utente del sistema
- **I casi d'uso** rappresentano una sequenza di interazioni per un tipo di funzionalità
- Il modello dei casi d'uso è l'insieme di tutti i casi d'uso. E' una descrizione completa della funzionalità del sistema e del suo ambiente

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

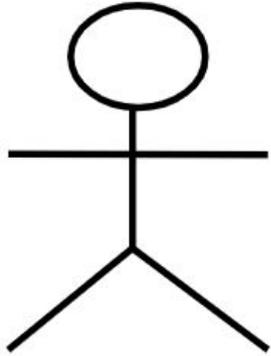
- Specificare il caso d'uso
 - con una descrizione testuale (narrativa)

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

Attori



Passeggero

Un attore modella un'entità esterna che comunica con il sistema:

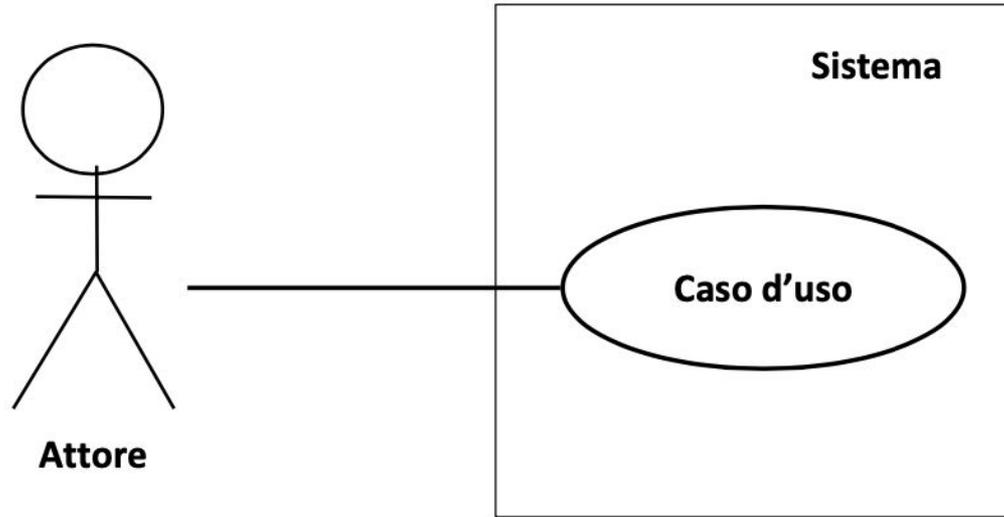
- Utente
- Sistema esterno
- Ambiente fisico

Un attore ha un nome unico e una descrizione opzionale

Esempi:

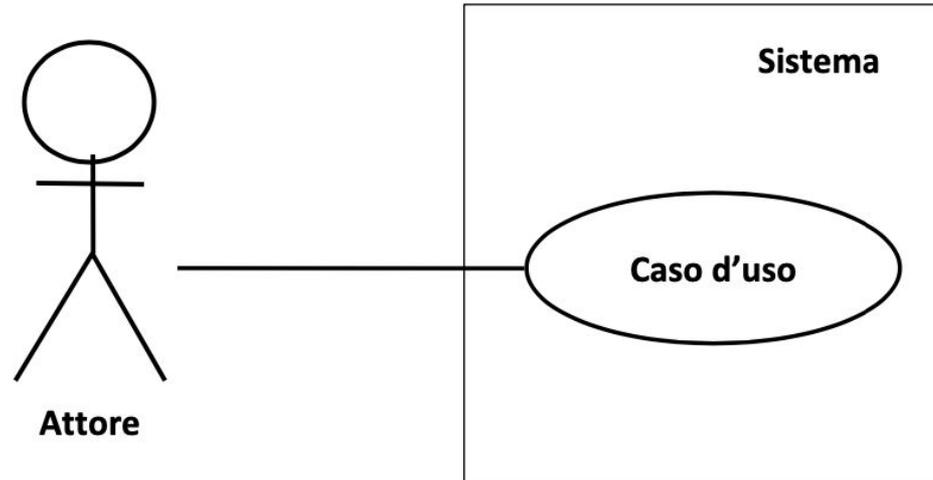
- Passeggero: Una persona nel treno
- Satellite GPS: fornisce al sistema le coordinate del GPS

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

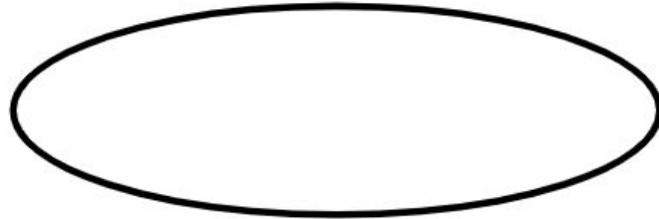
Semantica del caso d'uso



- 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**

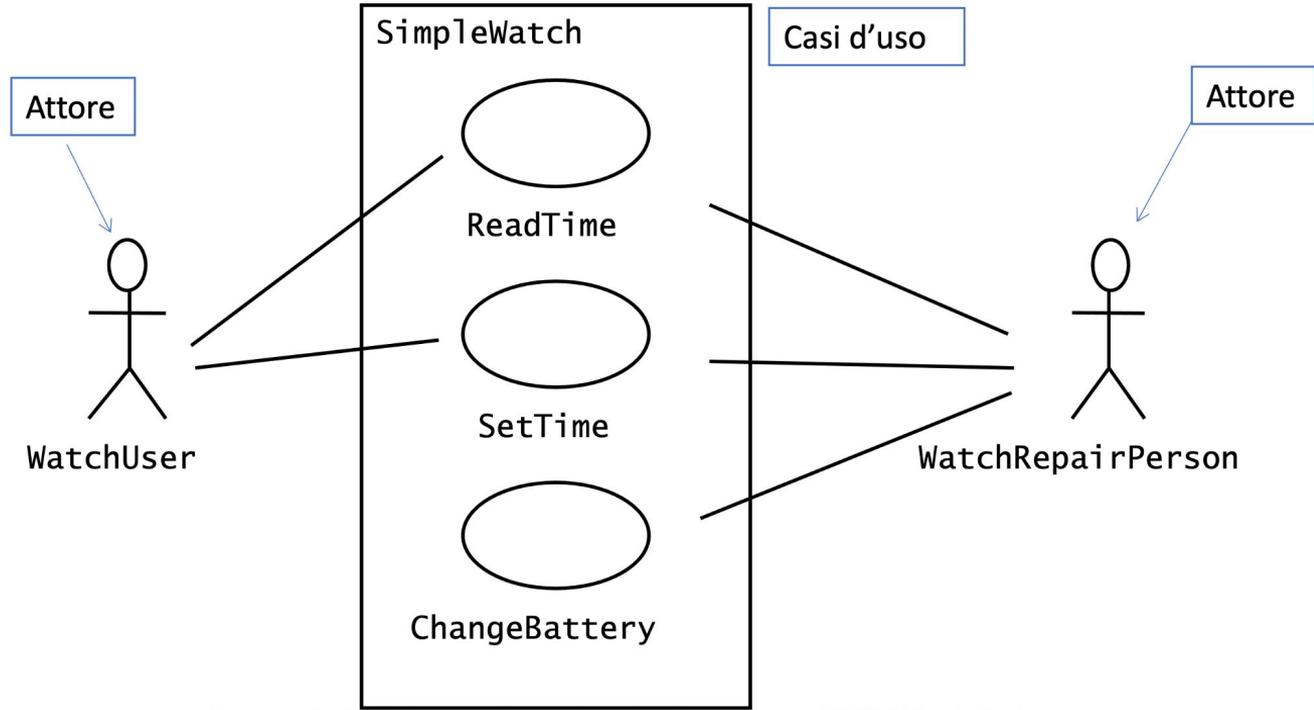
Casi d'uso

- Un caso d'uso rappresenta una classe di funzionalità fornite dal sistema
- I casi d'uso possono essere descritti testualmente, con attenzione sul flusso di eventi tra attore e sistema

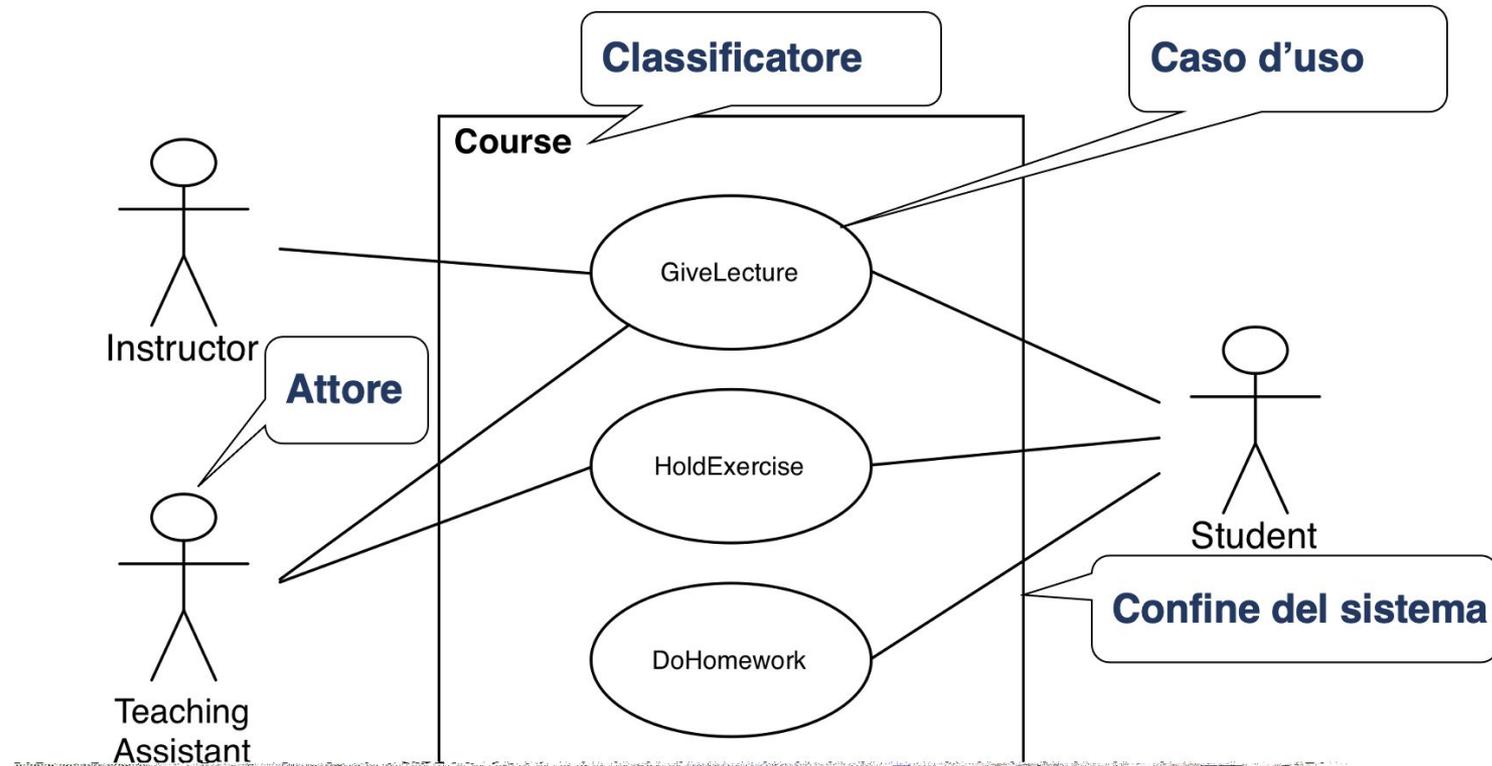


PurchaseTicket

Casi d'uso

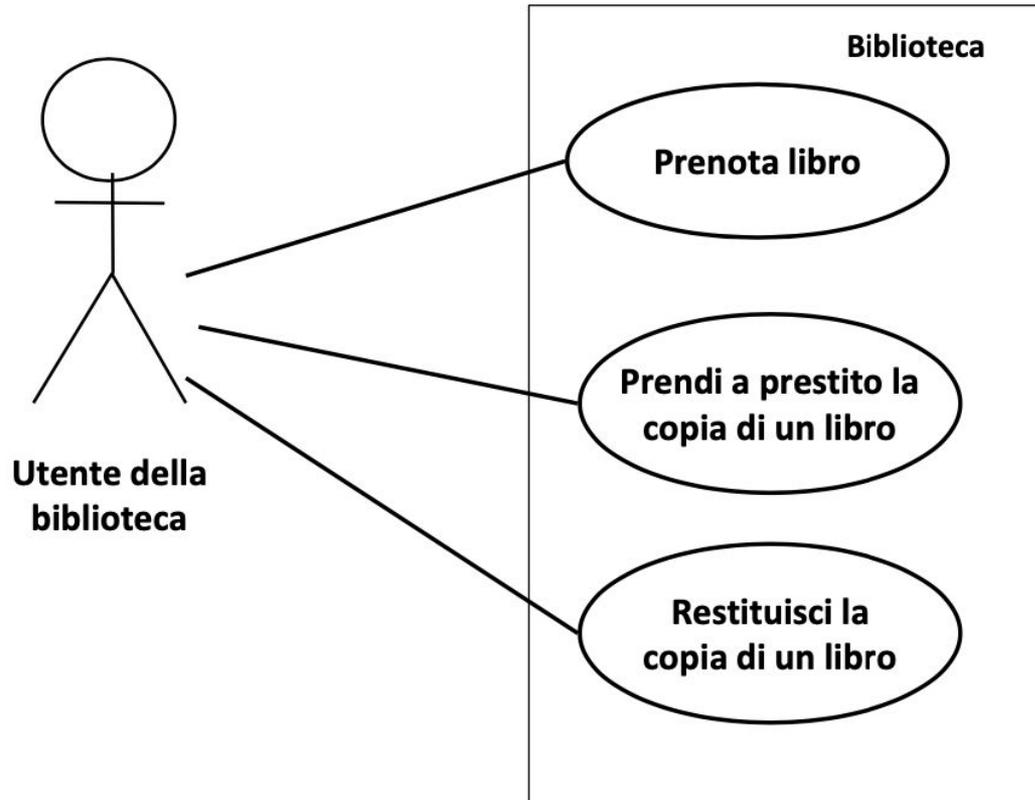


Esempio di diagramma dei casi d'uso



I diagrammi dei casi d'uso rappresentano le funzionalità del sistema dal punto di vista dell'utente

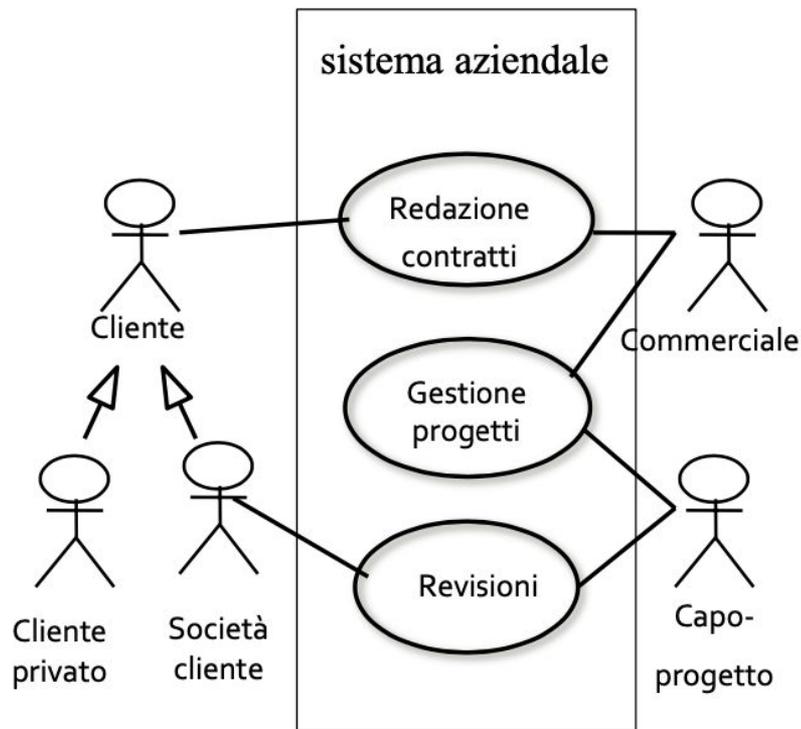
Esempio di diagramma dei casi d'uso



Esempio di diagramma dei casi d'uso

■ Esempio

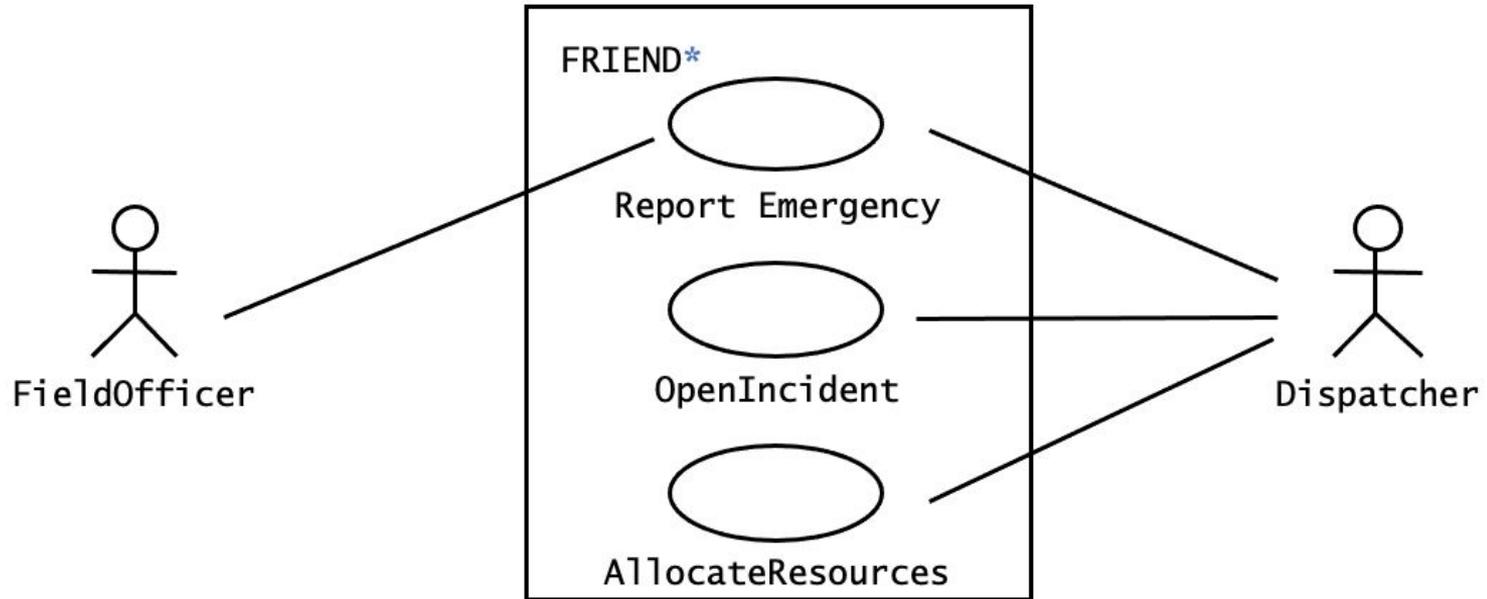
- I clienti possono essere privati o società
- Clienti e commerciali partecipano alla redazione dei contratti
- Società e capiprogetto partecipano alle revisioni
- Commerciali e capiprogetto gestiscono i progetti



Esempio: sistema gestione incidenti

- I **Field officer** (ad esempio, un poliziotto o un pompiere), hanno accesso ad un computer wireless che consente di interagire con un **Dispatcher**
 - Il dispatcher può visualizzare sul video del computer lo stato corrente di tutte le risorse, come le macchine di polizia o autocarri, e inviare una risorsa inserendo comandi da una stazione di lavoro
- In questo esempio FieldOfficer e Dispatcher sono attori

Esempio sistemi gestione degli incidenti



* First Responder Interactive Emergency Navigational Database

Descrizione testuale dei casi d'uso

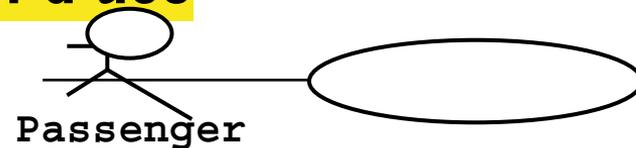
E' possibile descrivere testualmente un caso d'uso specificando:

1. Nome univoco
2. Attori partecipanti
3. Condizioni di ingresso
4. Condizioni di uscita
5. Flusso degli eventi
6. Requisiti speciali

Descrizione narrativa

<i>Nome:</i>	Nome del caso d'uso
<i>ID:</i>	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 testuale dei casi d'uso



1. Nome: PurchaseTicket

2. Attore partecipante:

Passenger

3. Condizione di ingresso

- Passenger si trova di fronte al Distributore di biglietti
- Passenger ha soldi sufficienti per acquistare il biglietto

4. Condizione di uscita:

Passenger ha il biglietto

5. Flusso degli eventi

1. Passenger seleziona l'area di destinazione (numero)
2. Il Distributore dei biglietti visualizza l'importo dovuto
3. Passenger inserisce i soldi, almeno l'importo dovuto
4. Il Distributore di biglietti restituisce il resto
5. Il Distributore di biglietti emette il biglietto

6. Requisiti speciali: nessuno

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">1. L'utente chiede di prolungare il prestito2. Il Sistema accerta che nessuno l'abbia prenotato3. Il Sistema accerta che l'utente sia ammesso al prestito4. Il Sistema aggiorna utente e stato del libro
<i>Postcondizioni:</i>	Prestito esteso, stato utente aggiornato
<i>Sequenze alternative degli eventi:</i>	Il libro è già prenotato, utente non ammesso al prestito

Template per i casi d'uso

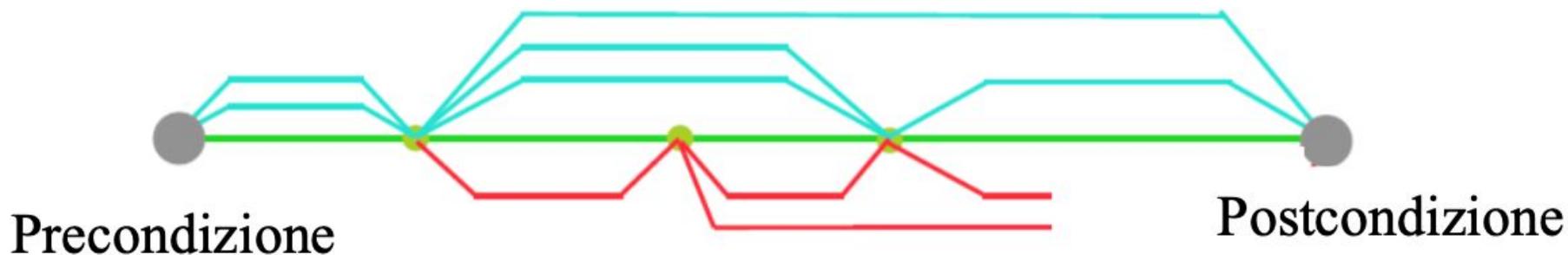
- Per descrivere un caso d'uso è possibile usare un template composto da sei campi
 - **Nome** del caso d'uso: unico in tutto il sistema in modo da essere usato senza ambiguità da tutti i partecipanti al progetto
 - **Attori partecipanti**: sono gli attori che interagiscono con il caso d'uso
 - **Condizioni di entrata**: descrivono le condizioni che devono essere soddisfatte prima che il caso d'uso sia iniziato
 - **Flusso di eventi**: descrive la sequenza di interazioni del caso d'uso. Sono numerati per riferimenti successivi
 - Per maggiore chiarezza, i casi comuni e i casi eccezionali sono descritti separatamente
 - **Condizioni di uscita**: descrive le condizioni che sono soddisfatte dopo il completamento del caso d'uso
 - **Requisiti di qualità**: requisiti non legati alla funzionalità. Vincoli sulle prestazioni del sistema

Nome	ReportEmergency
Attori Partecipanti	Iniziato dal FieldOfficer Comunica con il Dispatcher
Flusso eventi:	<ol style="list-style-type: none">1. Il FieldOfficer attiva la funzione “Report Emergency del terminale2. FRIEND risponde presentando una form al FieldOfficer3. Il FieldOfficer riempie la form selezionando il livello di emergenza, tipo, località, breve descrizione della situazione. Il FieldOfficer descrive anche possibili risposte alla situazione di emergenza. Appena finito Il FieldOfficer invia la form4. FRIEND riceve la form e notifica al Dispatcher5. Il Dispatcher rivede le informazioni sottomesse e crea un Incidente nel DB invocando il caso d’uso OpenIncident. Il Dispatcher seleziona una risposta e accetta il rapporto.6. FRIEND visualizza l’accettazione e la risposta selezionata al FieldOfficer
Condizioni di entrata	Il FieldOfficer è loggato in FRIEND
Condizioni di uscita	Il FieldOfficer ha ricevuto un’accettazione e una risposta selezionata dal Dispatcher, OR Il FieldOfficer ha ricevuto una spiegazione indicante perché la transazione non è stata eseguita
Requisiti di qualità	Il rapporto del FieldOfficer è accettato entro 30 secondi La risposta selezionata arriva non più tardi di 30 secondi dopo che è stata inviata dal Dispatcher

Scenari

- Un caso d'uso è un'astrazione che descrive tutti i possibili scenari che coinvolgono la funzionalità descritta
- Uno scenario è un'istanza del caso d'uso che descrive un insieme concreto di azioni
 - Sono usati come esempi per illustrare i casi comuni. L'obiettivo è la comprensibilità
 - I casi d'uso descrivono tutti i possibili casi. L'obiettivo è la completezza

Caso 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

Template per gli scenari

- Si usa un template con tre campi per descrivere uno scenario
 - Il **nome dello scenario**: consente di riferirlo in modo non ambiguo. Il nome è sottolineato per indicare che è un'istanza
 - Le **istanze degli attori partecipanti**: indica quali istanze degli attori sono coinvolte nello scenario
 - Anch'essi hanno nomi sottolineati
 - **Flusso di eventi**: descrive la sequenza di eventi passo dopo passo

Nome Scenario	<u>warehouseOnFire</u>
Istanze attori partecipanti	<u>bob, alice: FieldOfficer</u> <u>john: Dispatcher</u>
Flusso di eventi	<ol style="list-style-type: none"><li data-bbox="467 220 1740 345">1. Bob, guidando lungo la strada principale nella sua patrol, osserva del fumo proveniente da un magazzino. Il suo partner, Alice, attiva la funzione “Report Emergency” dal laptotp del FRIEND.<li data-bbox="467 355 1740 524">2. Alice immette l’indirizzo dell’edificio, una breve descrizione della sua ubicazione (angolo nord-ovest) ed un livello di emergenza. In aggiunta ad un’unità dei vigili del fuoco, richiede diverse unità paramediche poiché la zona non è molto trafficata. Conferma l’input ed attende l’accettazione.<li data-bbox="467 535 1740 703">3. John, il Dispatcher, è allertato per l’emergenza da un beep dalla sua stazione di lavoro. Rivede le informazioni sottomesse da Alice e accetta il rapporto. Alloca un’unità dei vigili del fuoco e due unità paramediche sul sito dell’Incidente ed invia la stima del tempo di arrivo (ETA) ad Alice.<li data-bbox="467 714 1039 751">4. Alice riceve l’accettazione e l’ETA.

Relazione tra i casi d'uso

- Quando i casi d'uso e gli attori si scambiano informazioni, si dice che essi comunicano
- Gli scambi di informazione possono essere rappresentati con le relazioni di comunicazione
- I diagrammi dei casi d'uso possono includere quattro tipi di relazioni: **comunicazione, inclusione, estensione ed ereditarietà**

Relazioni di comunicazione

- Attori e casi d'uso comunicano quando si scambiano informazioni
- Le relazioni di comunicazione sono illustrate con linee solide tra i simboli dell'attore e del caso d'uso
- In riferimento al sistema FRIEND di gestione incidenti
 - gli attori FieldOfficer e Dispatcher comunicano con il caso d'uso ReportEmergency
 - Solo l'attore Dispatcher comunica con i casi d'uso OpenIncident e AllocateResources
- Le relazioni di comunicazione tra attori e casi d'uso possono essere usate per denotare funzionalità di accesso
 - FieldOfficer e Dispatcher sono forniti di interfacce differenti al sistema ed hanno accesso a funzionalità differenti

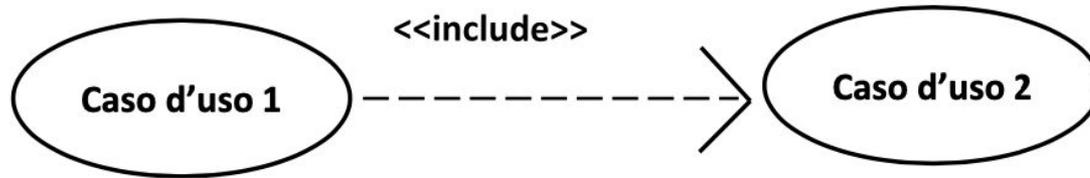
Relazioni di inclusione

- E' possibile ridurre la complessità di un modello identificando delle aspetti in comune tra differenti casi d'uso
 - Esempio: assumiamo che il Dispatcher premendo un tasto abbia accesso ad una mappa stradale
 - Ciò può essere modellato da un caso d'uso ViewMap **incluso** nei casi d'uso OpenIncident e AllocateResources
 - Il modello risultante descrive solo la funzionalità ViewMap una volta, riducendo di fatto la complessità del modello dei casi d'uso complessivo

Relazione di inclusione

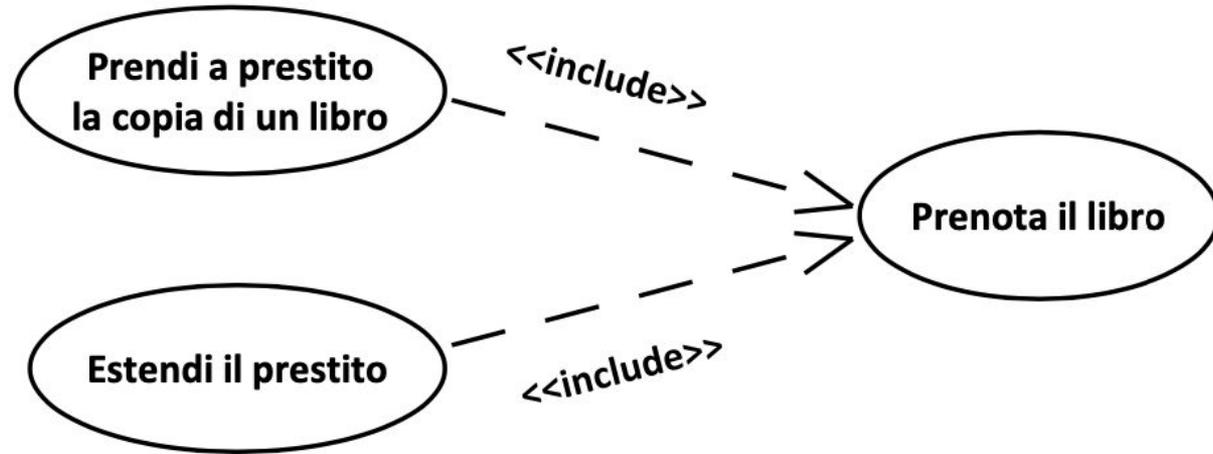
- Due casi d'uso sono legati da una relazione di inclusione se uno di essi include il secondo nel suo flusso di eventi
- In UML, le inclusioni sono rappresentate da frecce tratteggiate aperte che si originano dal caso d'uso che include
 - Le relazioni di inclusione sono etichettate con <<include>>

Inclusione di un 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

Esempio: relazione <include> testuale

Nome	AllocateResources
Attori Partecipanti	Iniziato da Dispatcher
Flusso eventi:	...
Condizioni di entrata	Il Dispatcher apre un Incidente
Condizioni di uscita	Sono assegnate Risorse aggiuntive ad un incidente Le risorse sono notificate dei nuovi assegnamenti Il FieldOfficer che ha a carico l'incidente è notificato delle nuove risorse
Requisiti di qualità	In qualsiasi punto del flusso di eventi, questo caso d'uso può includere il caso d'uso ViewMap. Il caso d'uso ViewMap è iniziato quando il Dispatcher invoca la funzione "map". Quando invocato all'interno di questo caso d'uso, il sistema scorre la mappa in modo che il luogo dell'incidente sia visibile al Dispatcher

Relazioni di estensione

- Le relazioni di estensione sono un modo alternativo per ridurre la complessità in un modello dei casi d'uso
- Un caso d'uso può estendere un altro caso d'uso aggiungendo eventi
- Una relazione di estensione indica che un caso d'uso estende un secondo caso d'uso (caso d'uso base) quando descrive in modo più ampio e dettagliato una variante dell'attività del caso d'uso base
- Una tipica applicazione di relazione di estensione è la specifica di un comportamento eccezionale

Estensione di un caso d'uso

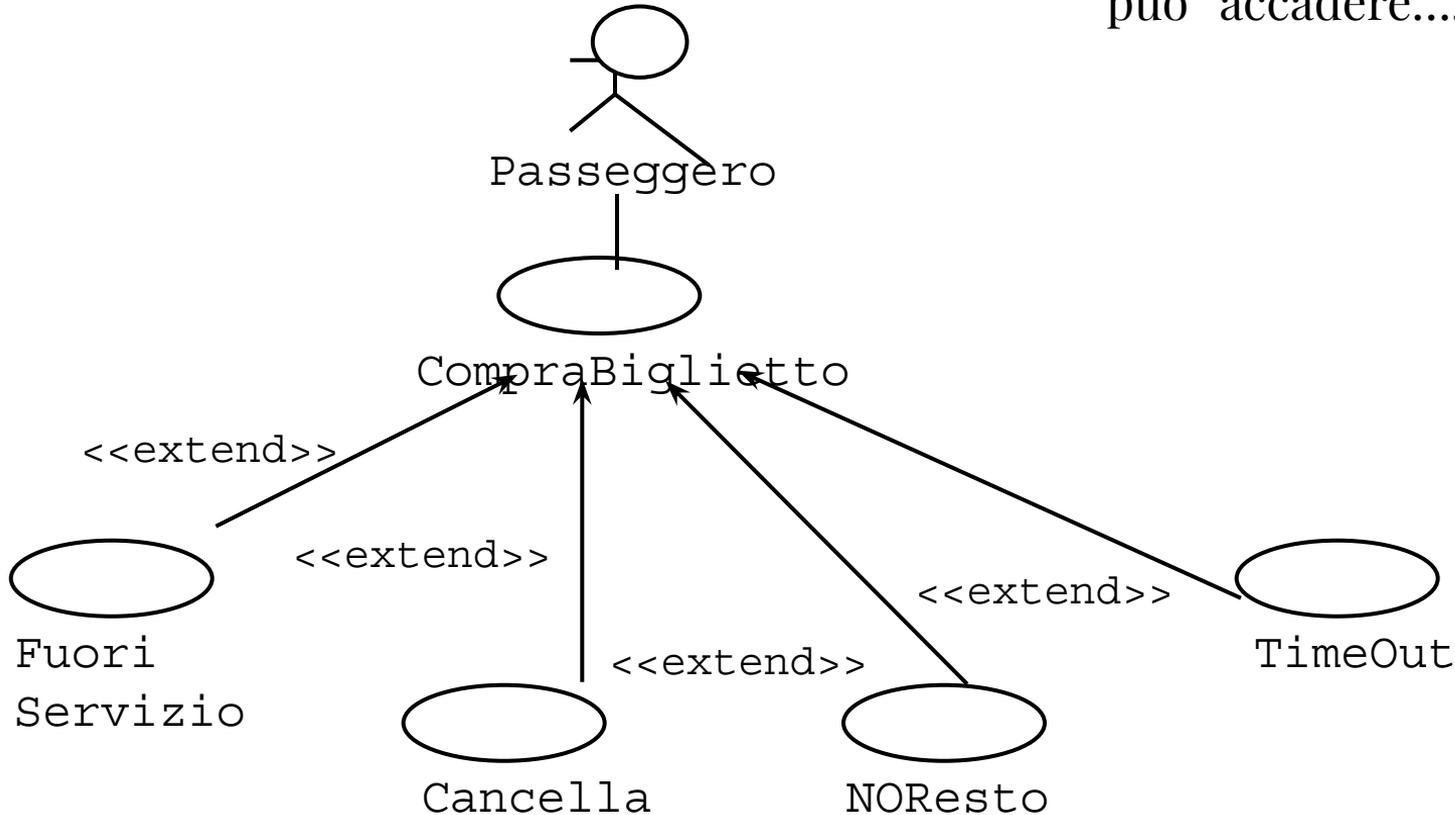


- La relazione extend punta al caso d'uso che viene esteso
 - Il caso d'uso 1 puo incorporare l'interazione descritta dal caso d'uso 2
 - La narrazione del caso 1 puo' in qualche caso invocare l'esecuzione del caso (2)
- <<extend>> è uno stereotipo
 - Gli stereotipi sono keywords che annotano elementi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

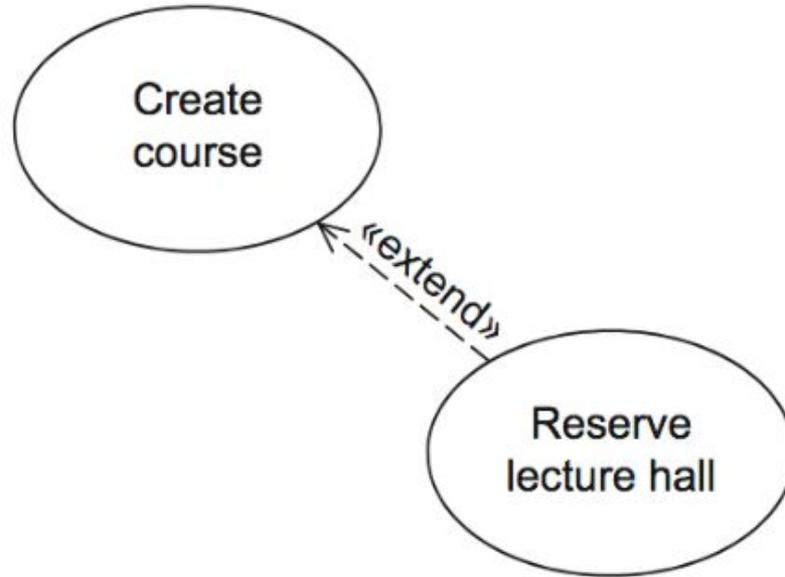
**Attenzione al verso della freccia
opposto all'include**

La relazione <<extend>>

Quando si acquista un biglietto
“può” accadere....



Esempio di estensione

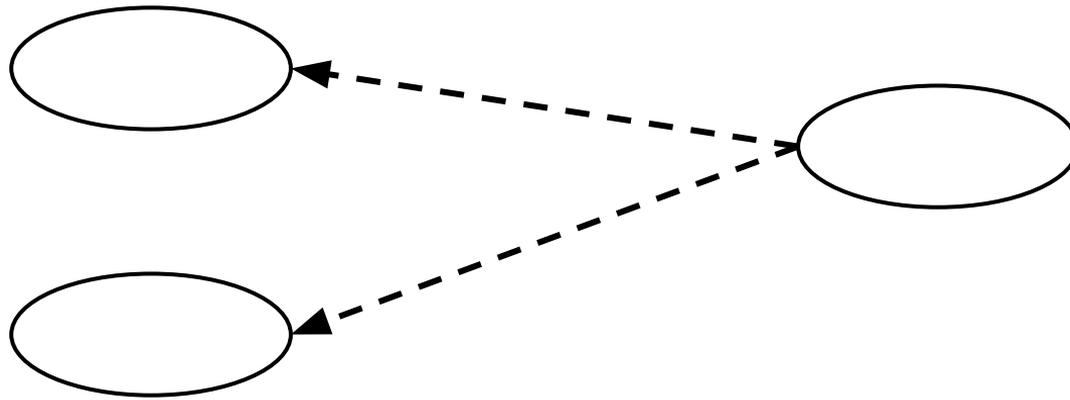


Quando si crea un nuovo corso **si può** riservare un'aula ma **non è obbligatorio**

Esempio

- Assumiamo che la connessione di rete tra il Dispatcher e il FieldOfficer si può interrompere in qualsiasi momento (ad esempio quando il FieldOfficer entra in un tunnel)
- Il caso d'uso ConnectionDown descrive l'insieme di eventi che riguardano il sistema e gli attori quando la connessione è persa
 - ConnectionDown estende i casi d'uso OpenIncident e AloccateResources
- Separare comportamenti eccezionali da comportamenti comuni consente di scrivere casi d'uso più brevi e maggiormente focalizzati

Esempio: Relazione <<extend>>



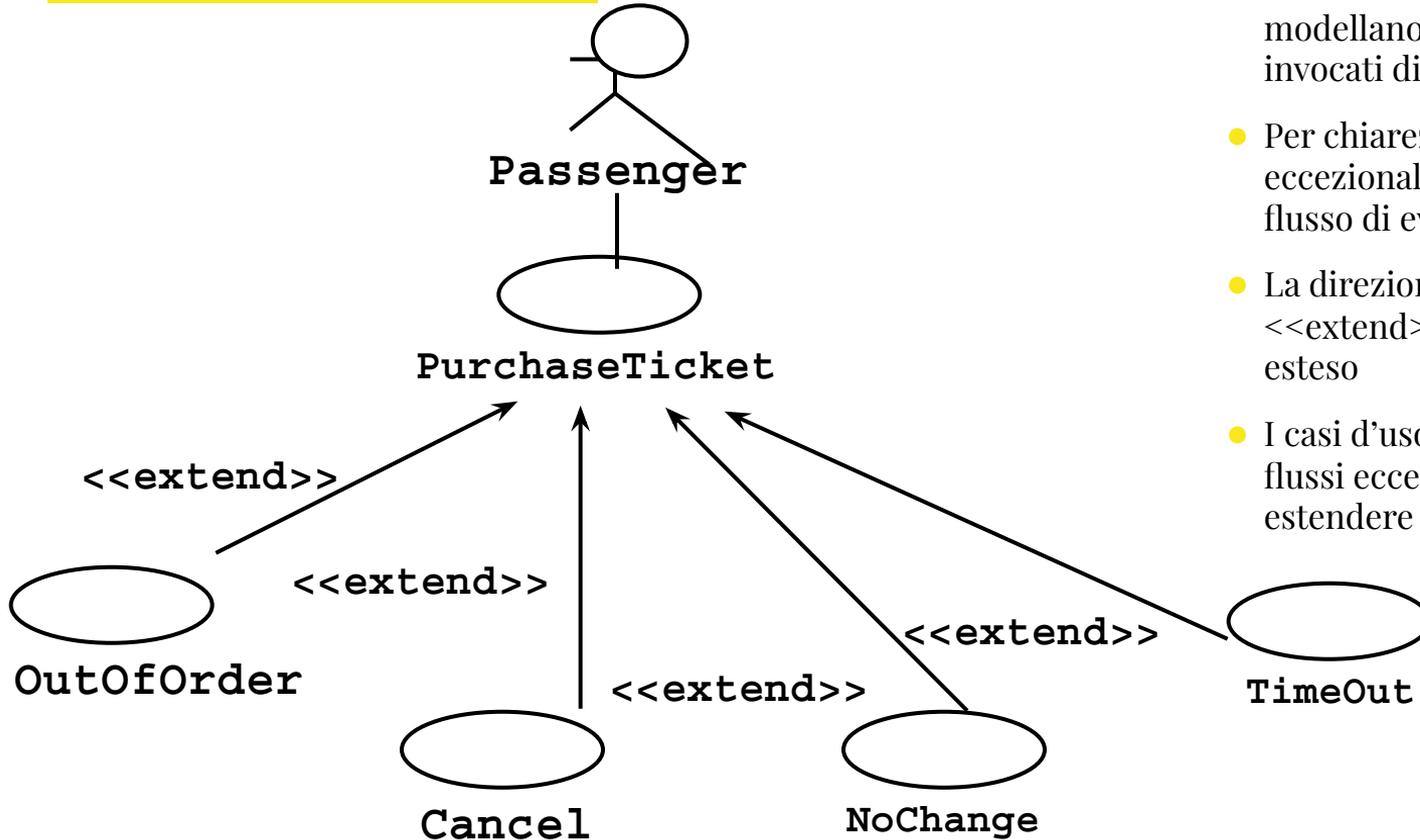
Esempio: Relazione <<extend>> testuale

Nome	ConnectionDown
Attori Partecipanti	Comunica con FieldOfficer e Dispatcher
Flusso eventi:	...
Condizioni di entrata	Questo caso d'uso estende i casi d'uso OpenIncident e AllocateResources. E' iniziato dal sistema quando la connessione di rete tra il FieldOfficer e il Dispatcher è persa.
Condizioni di uscita	...

Relazioni <<include>> ed <<extend>>

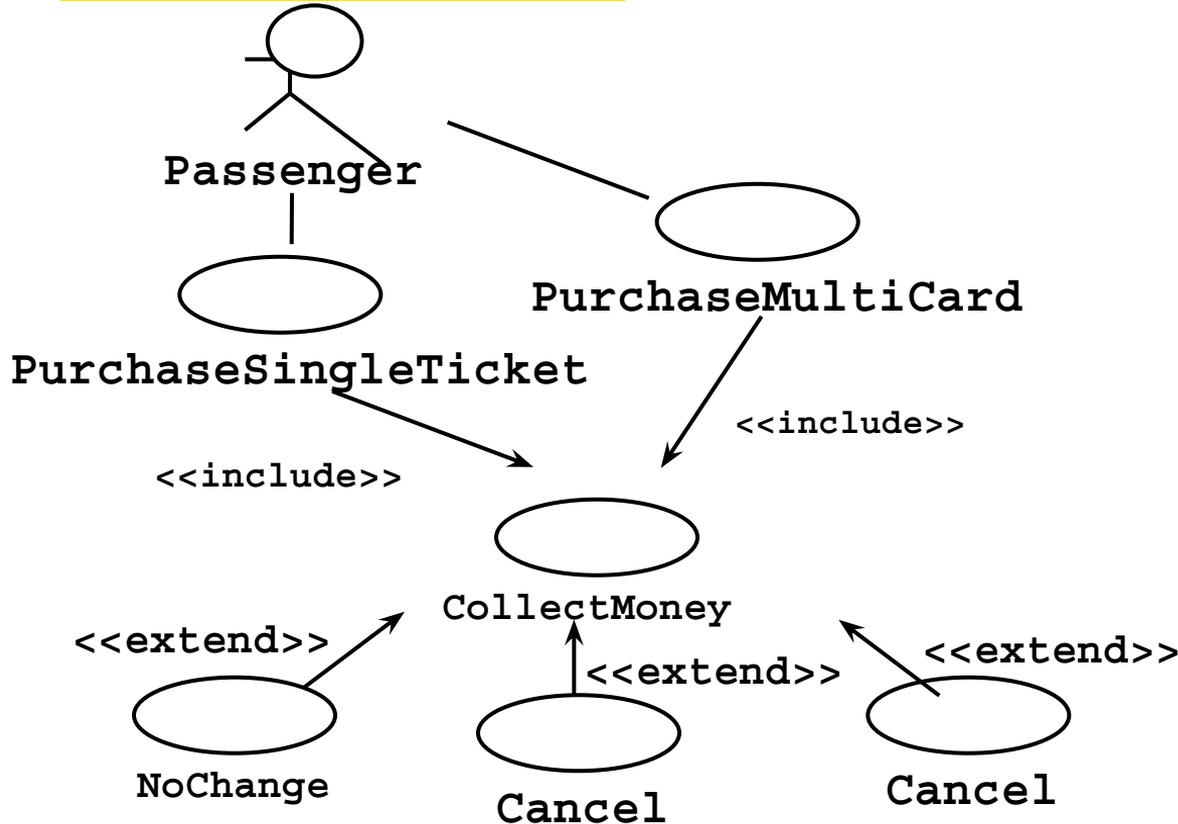
- La differenza tra le relazioni di inclusione e di estensione sta nel luogo della dipendenza
 - Assumiamo di aggiungere molti nuovi casi d'uso per l'attore Dispatcher, ad esempio UpdateIncident e ReallocateResources.
 - Se modellassimo il caso d'uso ConnectionDown con la relazione include, gli autori dei casi d'uso UpdateIncident e ReallocateResources dovrebbero conoscere ed includere il caso d'uso ConnectionDown
 - Con la relazione di estensione, invece, solo il caso d'uso ConnectionDown deve essere modificato per estendere i casi d'uso aggiuntivi
- In generale, i casi relativi ad eccezioni come help, errori e condizioni inattese, sono modellati con relazioni di estensione
- I casi d'uso che descrivono comportamenti condivisi da un insieme limitato di casi d'uso sono modellati con la relazione di inclusione

Relazione <<extend>>



- Le relazioni <<extend>> modellano casi eccezionali o invocati di rado
- Per chiarezza, i flussi di eventi eccezionali sono estrapolati dal flusso di eventi principale
- La direzione di una relazione <<extend>> è verso il caso d'uso esteso
- I casi d'uso che rappresentano i flussi eccezionali possono estendere più di un caso d'uso

Relazione <<include>>



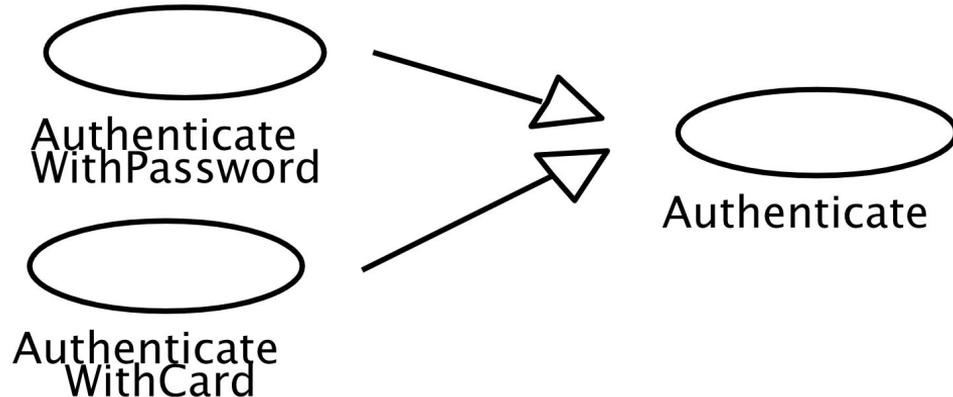
- La relazione <<include>> rappresenta funzionalità comuni necessarie in più di un caso d'uso
- Il comportamento di <<include>> è estrapolato per riuso e non perché rappresenta un'eccezione
- La direzione di una relazione <<include>> è verso il caso d'uso che usa (differentemente dalla direzione della relazione <<extend>>)

Relazioni di ereditarietà

- Terzo meccanismo per ridurre la complessità di un modello
- Un caso d'uso può specializzare un altro caso d'uso più generale aggiungendo più dettaglio
- Ad esempio, ai FieldOfficer è richiesto di autenticarsi prima di usare FRIEND
 - Durante le prime fasi della scoperta dei requisiti, l'autenticazione è modellata come un caso d'uso ad alto livello Authenticate
 - Successivamente, gli sviluppatori descrivono Authenticate con maggior dettaglio e consentono diverse piattaforme hardware
 - Il risultato sono due ulteriori casi d'uso, AuthenticateWithPassword, che consente ai FieldOfficer di autenticarsi senza hardware ausiliari e AuthenticateWithCard che consente di autenticarsi con una smart card

Relazione di ereditarietà

- I due nuovi casi d'uso sono rappresentati come specializzazioni del caso d'uso Authenticate
- Nella rappresentazione testuale, i casi d'uso specializzati ereditano l'attore che inizia e le condizioni di entrata e uscita dal caso d'uso generale



Esempio: relazione di ereditarietà testuale

Nome	AuthenticateWithCard
Attori Partecipanti	Ereditato dal caso d'uso Authenticate
Flusso eventi:	<ol style="list-style-type: none">1. Il FieldOfficer inserisce la propria carta nel Terminale2. Il terminale di campo accetta la carta e richiede all'attore il PIN3. Il FieldOfficer immette il PIN con il tastierino numerico4. Il Terminale controlla il PIN immesso con il PIN memorizzato sulla carta. Se i PIN sono uguali, il FieldOfficer è autenticato. Altrimenti, il Terminale rifiuta il tentativo di autenticazione
Condizioni di entrata	Ereditato dal caso d'uso Authenticate
Condizioni di uscita	Ereditato dal caso d'uso Authenticate

Extend e inheritance

- Le relazioni extend e inheritance sono differenti:
 - In extend ogni caso d'uso descrive un differente flusso di eventi per fare un compito differente
 - Il caso d'uso OpenIncident descrive le azioni che si verificano quando il Dispatcher crea un nuovo incidente, mentre il caso d'uso ConnectionDown descrive le azioni che occorrono durante l'interruzione della connessione
 - Nel caso della relazione di ereditarietà, invece, AuthenticateWithPassword e Authenticate entrambe descrivono le azioni che si verificano durante l'autenticazione, benché a differenti livelli di astrazione

Applicazione dei diagrammi dei casi d'uso

- I casi d'uso e gli attori definiscono i confini del sistema
- Sono sviluppati durante la fase di scoperta dei requisiti, spesso con i clienti e gli utenti
- Durante la specifica, i casi d'uso sono rifiniti e corretti poiché sono rivisti anche dagli sviluppatori e validati rispetto ai casi reali

ESERCIZIO 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.

