

Il diagramma dei casi d'uso

Francesco Poggi

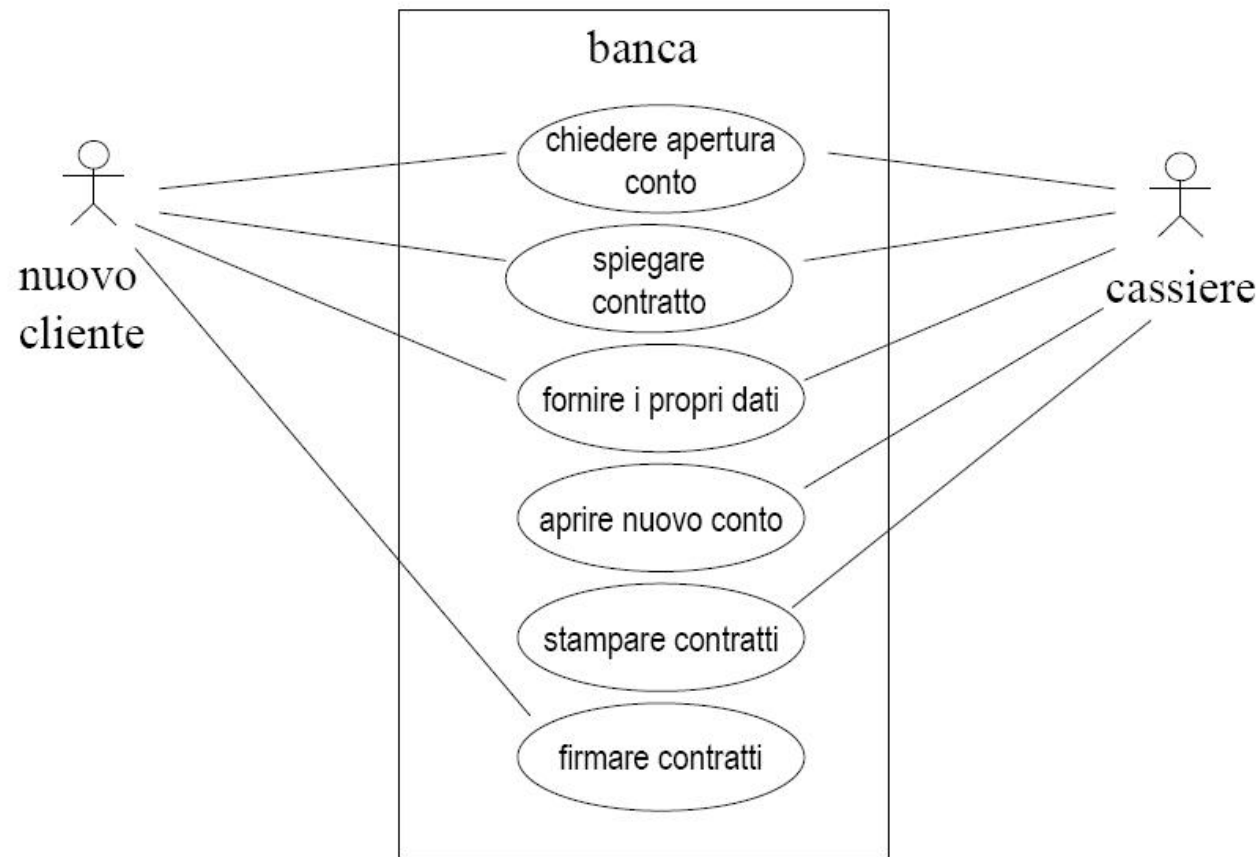
(dal materiale del Prof. Ciancarini, Dott. Favini e Di Iorio, Dott.ssa Zuppiroli)

A.A. 2015-2016

Il diagramma dei casi d'uso

- Si tratta di un diagramma che esprime un comportamento, desiderato o offerto.
- L'oggetto esaminato è solitamente un sistema o una sua parte
- Individua:
 - ▶ **chi** o che cosa ha a che fare con il sistema (**attore**)
 - ▶ **che cosa** l'attore può fare (**caso d'uso**).
- Tipicamente è il primo tipo di diagramma ad essere creato in un processo o ciclo di sviluppo, nell'ambito dell'analisi dei requisiti.

Un esempio: conto in banca



I casi d'uso

- I casi d'uso esistono da prima dei diagrammi UML!
- Proposti da Ivar Jacobson nel 1992 per il metodo Objectory
- In realtà la tecnica era già consolidata: *studio degli scenari di operatività degli utilizzatori di un sistema*
- In altre parole:
 - ▶ i modi in cui il sistema può essere utilizzato
 - ▶ le funzionalità che il sistema mette a disposizione dei suoi utilizzatori

Casi d'uso e requisiti funzionali

- I **requisiti funzionali** specificano **cosa** deve essere fatto.
- Sono indipendenti dalla tecnologia, dall'architettura, dalla piattaforma, dal linguaggio di programmazione.
- I requisiti **non-funzionali** specificano **vincoli aggiuntivi**, ad esempio:
 - ▶ performance
 - ▶ scalabilità
 - ▶ tolleranza ai guasti
 - ▶ dimensione degli eseguibili
- I **casi d'uso** modellano i **requisiti funzionali**.

Caso d'uso

- **Specifica cosa** ci si aspetta da un sistema
- **Nasconde come** il sistema lo implementa
- E' una sequenza di azioni (con varianti) che producono un risultato **osservabile da un attore**
- Si usa per
 - ▶ Descrivere i requisiti (analisi iniziale)
 - ▶ Convalidare l'architettura e verificare il sistema

Un passo indietro: i desiderata

- I *desiderata* sono ciò che il cliente desidera.
- Formalizzare i **desiderata** in **requisiti** è una delle più importanti sfide aperte dell'ingegneria del software.
- La specifica errata o incompleta delle richieste del cliente è una delle cause principali del fallimento dei progetti software.
- Il diagramma e la modellazione dei casi d'uso aiutano l'interazione con il cliente e migliorano l'estrazione dei requisiti (funzionali).

Esempio di desiderata: vanno bene?

Cliente:

- vorrei vendere i manufatti che realizzo...
- non vorrei solo un mercato locale...
- mi piacerebbe che gli acquirenti potessero visionare un catalogo da cui scegliere...
- vorrei gestire gli ordini da qualunque posto perché viaggio molto...

Che cosa viene fatto? Da chi?

Riformulati meglio...

Cliente:

- vorrei avere la possibilità di creare un catalogo dei miei manufatti...
- vorrei un catalogo liberamente consultabile da chiunque...
- vorrei organizzare i manufatti raccogliendoli in categorie...
- vorrei che gli interessati all'acquisto potessero inviarmi un ordine, che io provvederò ad evadere previa una qualche forma di registrazione...

Estrarre i requisiti

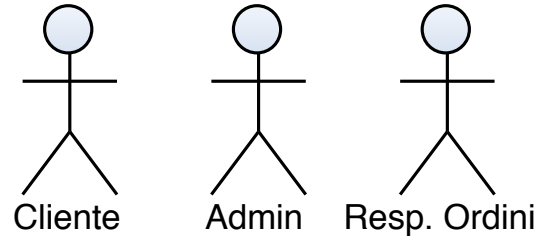
Chi interagisce con il sistema (**attori**)?

- Clienti
- Amministratori del negozio online
- Reparto ordini

Cosa fanno (**casi d'uso**)?

- Il cliente si registra, consulta il catalogo ed effettua acquisti
- L'amministratore organizza il catalogo, che è diviso in categorie
- Il reparto ordini riceve ordini da evadere
- Il cliente sceglie il tipo di pagamento

Attori



- Un attore specifica un ruolo assunto da un utente o altra entità che interagisce col sistema nell'ambito di un'unità di funzionamento (caso d'uso).
- Un attore è **esterno** al sistema.
- **Non è necessariamente umano**: oggetto fisico, agente software, condizioni ambientali, etc.
- Gli attori eseguono casi d'uso: prima si cercano gli attori, poi i loro casi d'uso!

Come individuare gli attori

- Per individuare un attore è necessario individuare chi/cosa interagisce col sistema e con quale ruolo.
- Domande utili:
 - ▶ Chi/cosa usa il sistema?
 - ▶ Che ruolo ha chi/cosa interagisce col sistema?
 - ▶ In quale parte dell'organizzazione è utilizzato il sistema?
 - ▶ Chi/cosa avvia il sistema?
 - ▶ Chi supporterà e manterrà il sistema?
 - ▶ Altri sistemi interagiscono col sistema?
 - ▶ Ci sono funzioni attivate periodicamente? *es. backup*
 - ▶ Chi/cosa ottiene o fornisce informazioni dal sistema?
 - ▶ Un attore ha diversi ruoli? Lo stesso ruolo è assegnato a più attori?

Caso d'uso UML (1)



- *La specifica di una sequenza di azioni, incluse eventuali sequenze alternative e/o di errore che un sistema (o sottosistema) può eseguire interagendo con attori esterni.*
- È qualcosa che un attore vuole il sistema faccia:
 - ▶ È descritto dal punto di vista dell'attore
 - ▶ È causato da un'azione di un attore
- Il nome (etichetta) dovrebbe essere basato su un verbo o su un sostantivo che esprime un avvenimento.

Caso d'uso UML (2)

- Un caso d'uso è sempre iniziato da un attore: in UML, un evento è sempre legato all'entità che lo ha generato.
- L'attore che inizia un caso d'uso è detto *primario*, gli altri attori che interagiscono nell'ambito di quel caso d'uso sono *secondari*.
- Un caso d'uso è un *classificatore dotato di comportamento*: può essere specificato da diagrammi di stato, interazione, sequenza, avere pre- e post-condizioni, ...
- Può ammettere al suo interno *varianti* al comportamento principale.

Come individuare un caso d'uso

- Individuare i casi d'uso è un'operazione **iterativa**.
- Per individuare i casi d'uso possono essere utili le seguenti domande:
 - ▶ Ciascun attore che funzioni si aspetta?
 - ▶ Il sistema gestisce (archivia/fornisce) informazioni? Se sì quali sono gli attori che provocano questo comportamento?
 - ▶ Alcuni attori vengono informati quando il sistema cambia stato?
 - ▶ Gli attori devono informare il sistema di cambiamenti improvvisi?
 - ▶ Alcuni eventi esterni producono effetti sul sistema?
 - ▶ Quali casi d'uso mantengono il sistema?
 - ▶ I requisiti funzionali sono tutti coperti dai casi d'uso?

Come recuperare informazioni su un caso d'uso

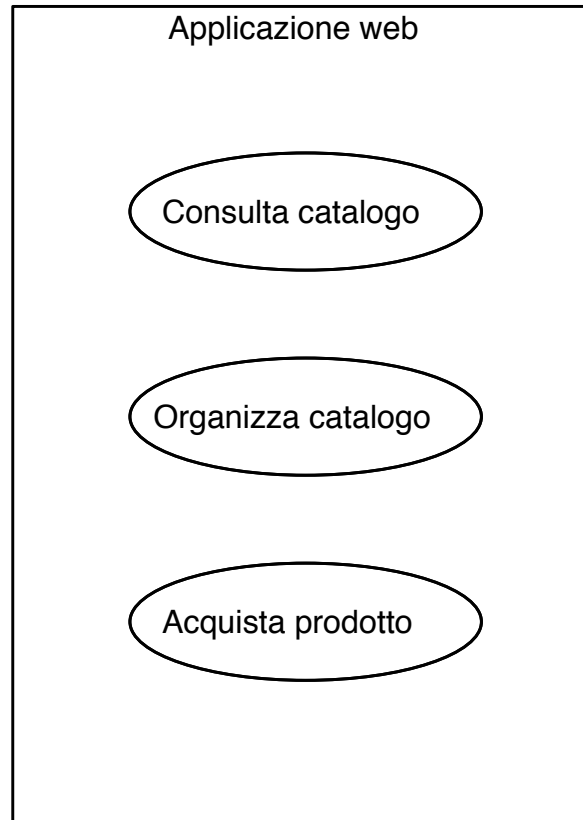
- Interviste con gli esperti del dominio (*desiderata* ben strutturati)
- Bibliografia del dominio del sistema
- Sistemi già esistenti
- Conoscenza personale del dominio

Come descrivere un caso d'uso

- Descrivere in modo generico e sequenziale il flusso di eventi di un caso d'uso
 - ▶ Descrivere la preconditione (stato iniziale del sistema)
 - ▶ Elencare la sequenza di passi
- Descrivere le interazioni con gli attori e i messaggi scambiati
- Aggiungere eventuali punti di estensione
- Descrivere in modo chiaro, preciso e breve

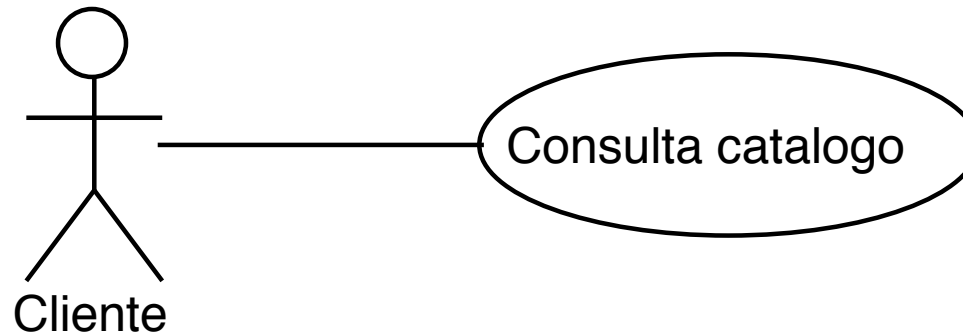
[ci torneremo più avanti]

Elementi del diagramma: sistema



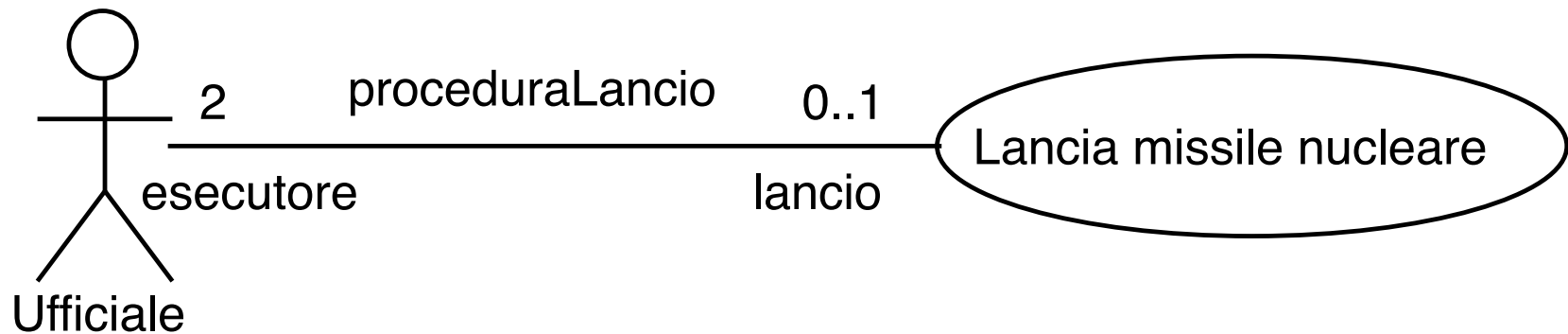
- Delimita l'argomento del diagramma, specificando i confini del sistema.

Elementi del diagramma: associazione (1)



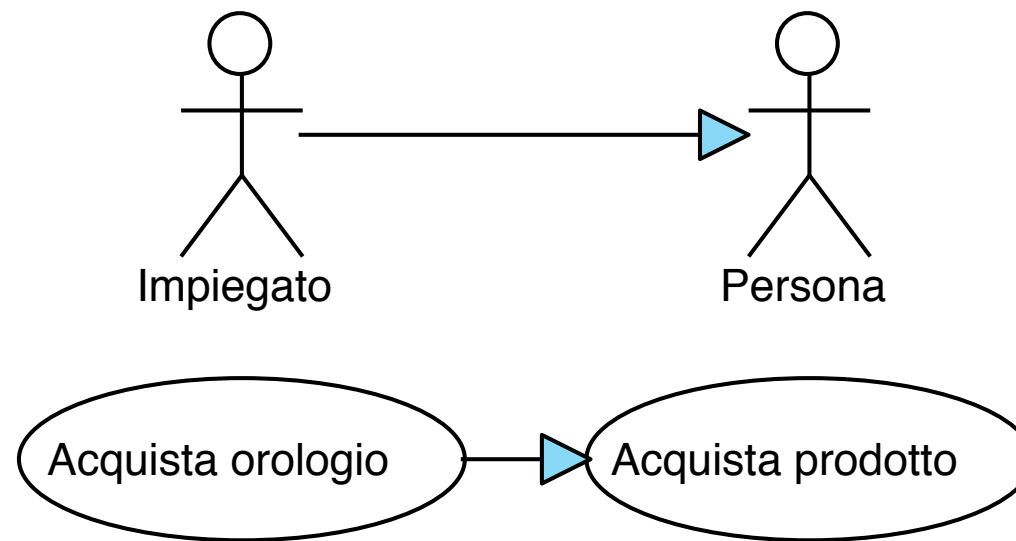
- Collega gli attori ai casi d'uso.
- Un attore si può associare solo a casi d'uso, classi e componenti (che verranno eventualmente associati ad altri casi d'uso, classi e componenti).
- Un caso d'uso non dovrebbe essere associato ad altri casi d'uso riguardanti lo stesso argomento.

Elementi del diagramma: associazione (2)



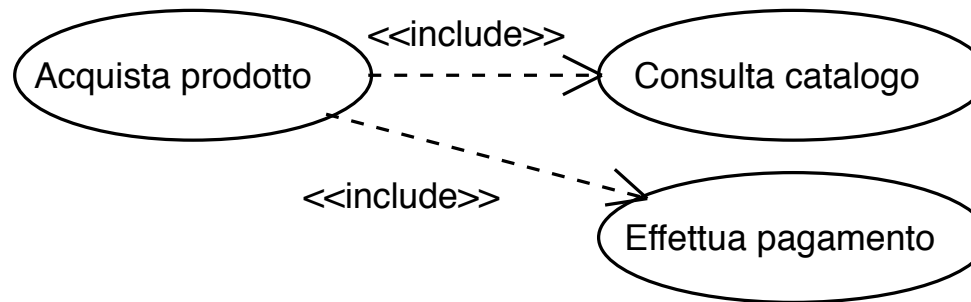
- Alcune caratteristiche opzionali comuni a tutte le associazioni in UML:
 - ▶ nome
 - ▶ molteplicità
 - ▶ ruoli

Elementi del diagramma: generalizzazione



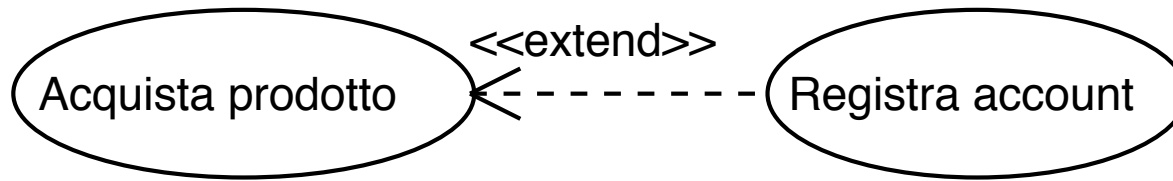
- Collega un attore o caso d'uso ad un altro più generale.
- Il figlio può sostituire il genitore dovunque questi appaia.

Elementi del diagramma: include



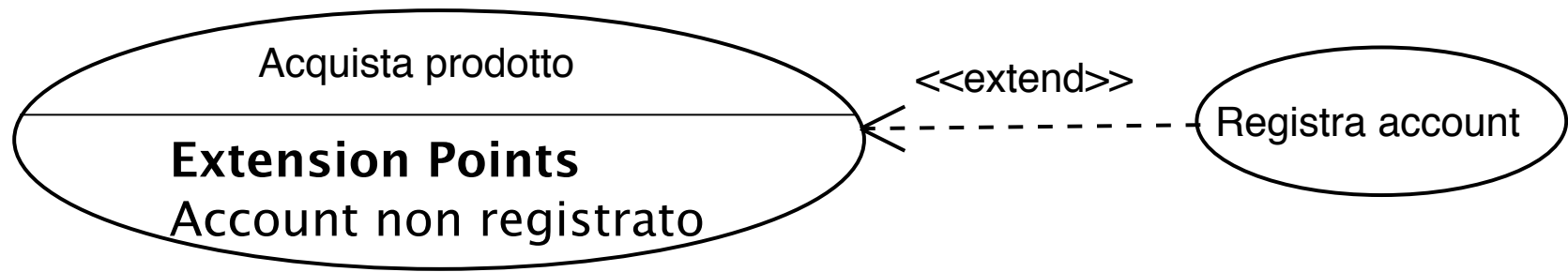
- Una **dipendenza** tra casi d'uso; il caso incluso fa parte del comportamento di quello che lo include.
- L'inclusione **non** è **opzionale** ed avviene in ogni istanza del caso d'uso.
- La corretta esecuzione del caso d'uso che include dipende da quella del caso d'uso incluso.
- Non si possono formare cicli di include.
- Usato per **riutilizzare parti comuni** a più casi d'uso.

Elementi del diagramma: extend



- Una **dipendenza** tra casi d'uso (notare il verso della freccia).
- Il caso d'uso che estende (client) specifica un incremento di comportamento a quello esteso (supplier).
- Si tratta di comportamento **supplementare** ed **opzionale** che gestisce **casi particolari** o non standard.
- *Diverso* da una generalizzazione tra casi d'uso:
 - ▶ in una generalizzazione, entrambi i casi d'uso sono ugualmente significativi
 - ▶ in un extend, il client non ha necessariamente senso se preso da solo

Extension points

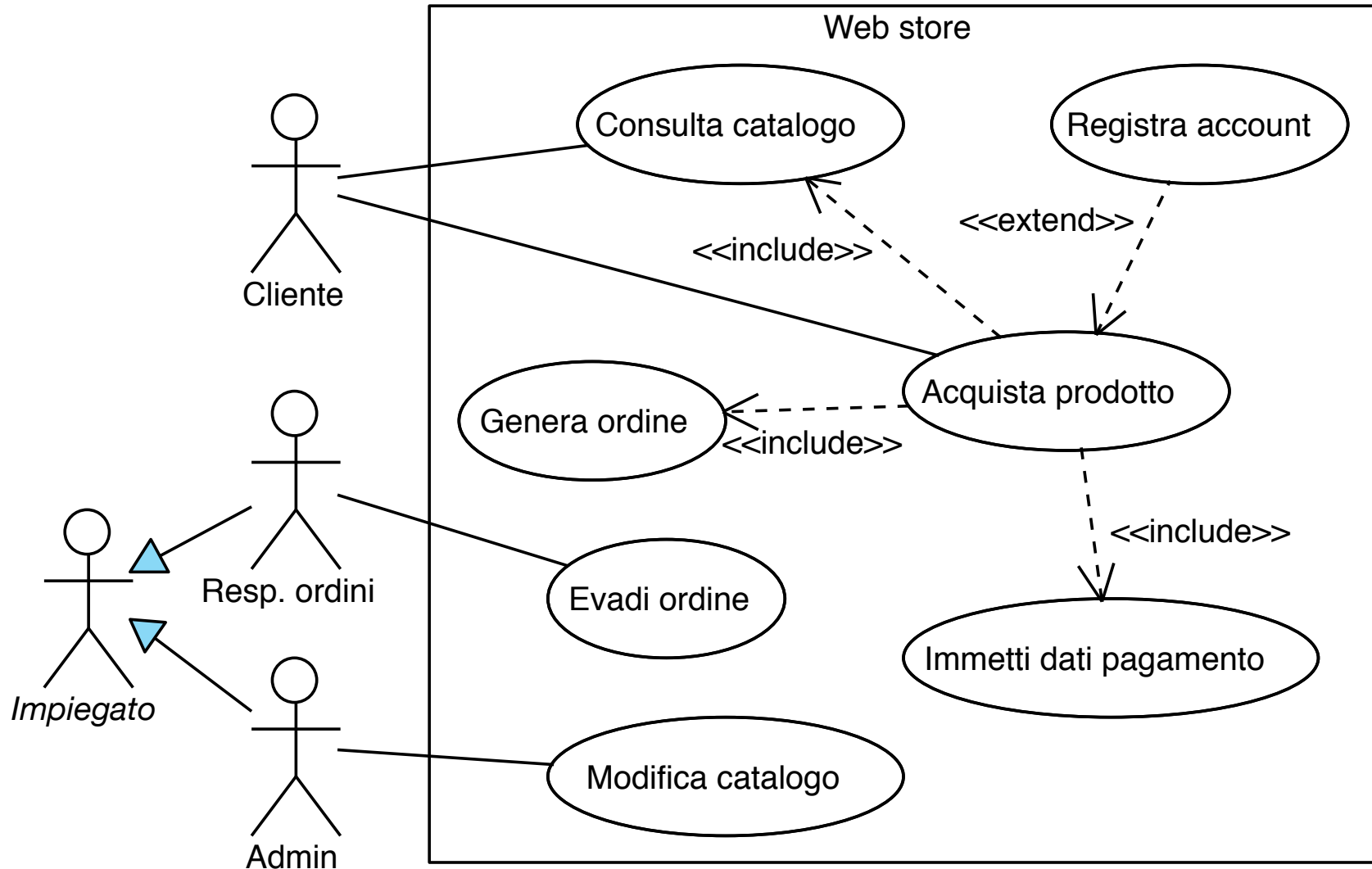


- Un caso d'uso raggiunto da almeno un extend può opzionalmente visualizzare i propri extension points.
- Specifica i **punti** e/o **condizioni** dell'esecuzione in cui il comportamento viene esteso.
- Se gli extension points sono molti, alcuni tool possono supportare la rappresentazione a rettangolo (i casi d'uso sono classificatori).

include vs. extend

- Include specifica comportamento **obbligatorio**.
- Extend specifica comportamento **supplementare** (varianti).
- Nell'include la freccia va dal caso d'uso che include verso quello incluso.
- Nell'extend la freccia va dal caso d'uso che estende verso quello esteso.
- Sono entrambi costrutti utili, ma non se ne deve abusare, o la leggibilità ne risente.

Esempio diagramma



Sono sufficienti questi diagrammi?

- Spesso nasce l'esigenza di abbinare i diagrammi dei casi d'uso a **specifiche testuali** più formali.
- I diagrammi dei casi d'uso non sono adatti a mostrare:
 - ▶ la sequenza temporale delle operazioni
 - ▶ lo stato del sistema e degli attori prima e dopo l'esecuzione del caso d'uso
- Manca la parte più importante: la descrizione (specificata) dei casi d'uso!

Specifiche del caso d'uso

- Ogni caso d'uso ha un **nome** e una **specifica**.
- La specifica è composta da:
 - ▶ **precondizioni**: condizioni che devono essere vere prima che il caso d'uso si possa eseguire
 - ▶ **sequenza degli eventi**: i passi che compongono il caso d'uso
 - ▶ **postcondizioni**: condizioni che devono essere vere quando il caso d'uso termina l'esecuzione

Esempio specifica caso d'uso

<i>Nome del caso d'uso</i>	Caso d'uso: Pagamento IVA
<i>Identificatore univoco</i>	ID: UC1
<i>Gli attori interessati dal caso d'uso</i>	Attori: Tempo, Fisco
<i>Lo stato del sistema prima che il caso d'uso possa iniziare</i>	Precondizioni: 1. Si è concluso un trimestre fiscale
<i>I passi del caso d'uso</i>	Sequenza degli eventi: 1. Il caso d'uso inizia quando si conclude un trimestre fiscale. 2. Il sistema calcola l'ammontare dell'IVA dovuta al Fisco. 3. Il sistema trasmette un pagamento elettronico al Fisco.
<i>Lo stato del sistema quando l'esecuzione del caso d'uso è terminata</i>	Postcondizioni: 1. Il Fisco riceve l'importo IVA dovuto.

Sequenza degli eventi

- Un **elenco di azioni** che definisce il caso d'uso nella sua **completezza**.
- Il caso d'uso si considera eseguito solo se l'esecuzione arriva fino alla fine.
- Un'azione è sempre iniziata da un attore oppure dal sistema (in UML, gli eventi sono sempre legati a chi li crea).
 - ▶ **Passo iniziale:** 1. Il caso d'uso inizia quando `<attore> <azione>...`
 - ▶ **Passi successivi:** `<numero>.` Il `<attore/sistema>` `<azione>`

Sequenza (corretta) di eventi

- ~~Incomincia quando si seleziona la funzione 'ordina libro'~~
- Il caso d'uso inizia quando il cliente seleziona la funzione 'ordina libro'
- ~~Vengono inseriti i dati del cliente~~
- Il cliente inserisce nel form il suo nome e indirizzo
- Il sistema verifica i dati del Cliente

Flusso degli eventi

- Ogni caso d'uso:
 - ▶ Ha una sequenza di transizioni (eventi) normale o di base
 - ▶ Può avere varie sequenze alternative
 - ▶ Ha varie sequenze di transazioni eccezionali per la gestione di errori o casi particolari
- Per descrivere correttamente il flusso di eventi quindi:
 - ▶ Descrive solo gli eventi relativi al caso d'uso, e non quel che avviene in altri casi d'uso
 - ▶ Descrivere come e quando il caso d'uso inizia e finisce
 - ▶ Descrivere il flusso base degli eventi
 - ▶ Descrivere ogni flusso alternativo

E in UML?

- UML usa parole chiave per esprimere queste variazioni alla sequenza principale.
- Parola chiave **Se**: indica una ramificazione della sequenza degli eventi nel caso si verifichi una condizione.
- Ripetizioni all'interno di una sequenza:
 - ▶ parola chiave **Per** (For)
 - ▶ parola chiave **Fintantoché** (While)
- È bene non eccedere con le ramificazioni!

Ramificazioni e sequenze alternative

- Facciamo un esempio partendo dal caso d'uso 'aggiorna carrello' di un negozio on-line.
- Possibili ramificazioni, dopo aver aggiunto un articolo al carrello:
 - ▶ se il cliente richiede una nuova quantità il sistema aggiorna la quantità di quell'articolo
 - ▶ se il client seleziona rimuovi articolo il sistema elimina quell'articolo dal carrello
- Le sequenze alternative sono invece ramificazioni che non possono essere espresse utilizzando il **Se**.
 - ▶ Ad esempio dovute a condizioni che si possono verificare in un qualunque momento: il cliente abbandona la pagina del carrello

Esempio ramificazioni

Caso d'uso: AggiornaCarrello
ID: UC2
Attori: Cliente
Precondizioni: 1. Il contenuto del carrello è visibile
Sequenza degli eventi: 1. Il caso d'uso inizia quando il Cliente seleziona un articolo nel carrello. 2. Se il Cliente seleziona “rimuovi articolo” 2.1 Il Sistema elimina l'articolo dal carrello. 3. Se il Cliente digita una nuova quantità 3.1 Il Sistema aggiorna la quantità dell'articolo presente nel carrello
Postcondizioni: 1. Il contenuto del carrello è stato aggiornato
Sequenza alternativa 1: 1. In qualunque momento il Cliente può abbandonare la pagina del carrello
Postcondizioni:

Come individuare sequenze alternative?

- Ad ogni passo della sequenza degli eventi principale, cercare:
 - ▶ alternative all'azione eseguita in quel passo
 - ▶ errori possibili nella sequenza principale
 - ▶ interruzioni che possono avvenire in qualunque momento della sequenza principale
- Sequenze alternative abbastanza complesse possono essere descritte separatamente.
 - ▶ stessa sintassi, si sostituisce 'caso d'uso' con 'sequenza degli eventi alternativa'
 - ▶ il primo passo può indicare il punto della sequenza principale da cui si proviene

Scenari

- Uno **scenario** rappresenta una particolare interazione tra uno o più attori e il sistema.
- Non contiene ramificazioni o sequenze alternative: è semplicemente la **cronaca di un'interazione** vera o verosimile.
 - ▶ il concetto risulta più immediato pensando agli attori in maniera concreta: non un generico cliente, ma Alice o Bob
- Una definizione alternativa: un caso d'uso è un insieme di scenari possibili se un utente prova a raggiungere un dato risultato

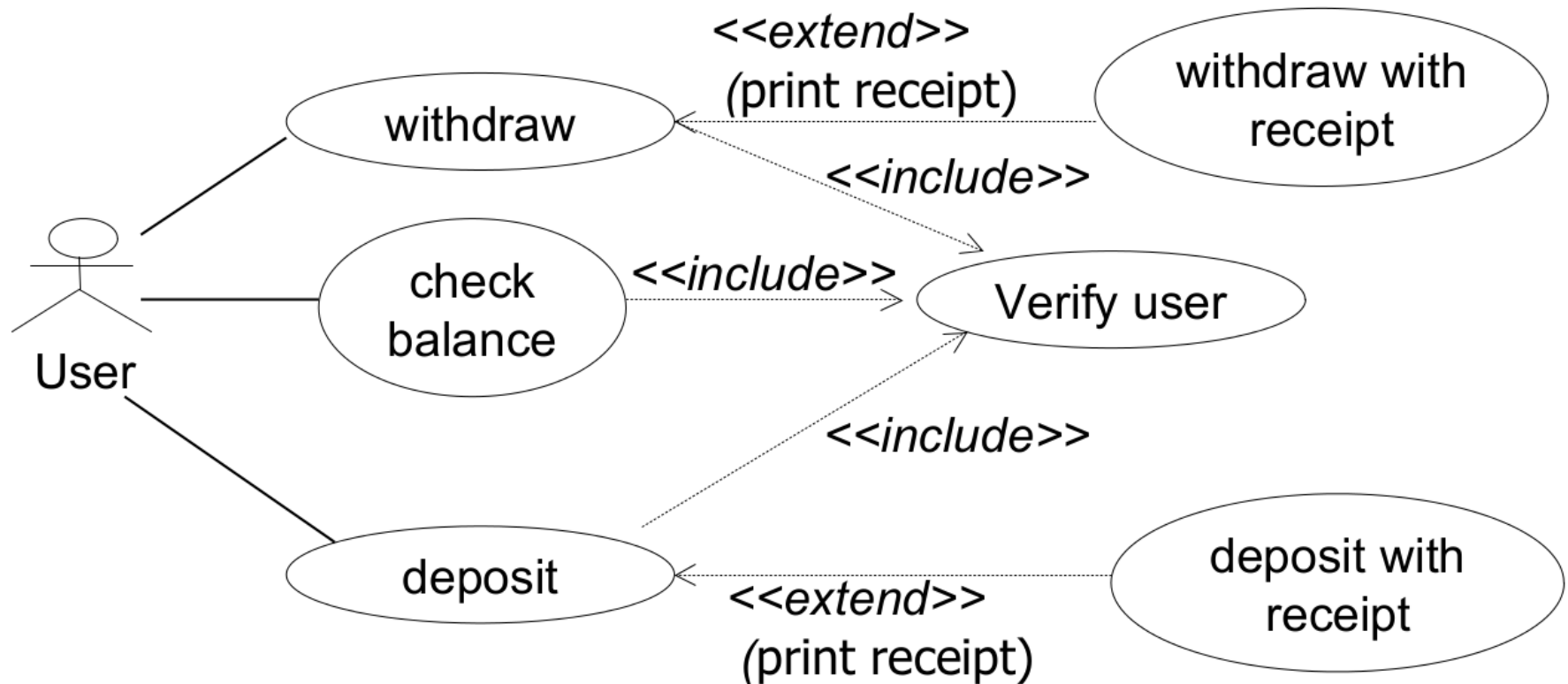
Scenario principale e scenari secondari

- Corrispondono ad esecuzioni della **sequenza principale** e di quelle **alternative**, rispettivamente.
- Strategie per limitare il numero, potenzialmente enorme, degli scenari secondari:
 - ▶ documentare solo quelli considerati più importanti
 - ▶ se ci sono scenari secondari molto simili, se ne documenta uno solo, se necessario aggiungendo annotazioni per spiegare come gli altri scenari differiscano dall'esempio.

Un esempio completo: conto in banca

- Si consideri il seguente sistema:
 - ▶ Il sistema usa uno sportello Bancomat
 - ▶ L'utente deve poter depositare assegni
 - ▶ L'utente deve poter ritirare contante
 - ▶ L'utente deve poter chiedere il saldo
 - ▶ L'utente deve poter ottenere una ricevuta se lo richiede. La ricevuta riporta il tipo di transazione, la data, il numero di conto, la somma, ed il nuovo saldo
 - ▶ Dopo ciascuna transazione viene visualizzato il nuovo saldo
- Disegnare il diagramma dei casi d'uso
- Scrivere la specifica dei principali casi d'uso

Soluzione



Soluzione

Use case: withdraw

- Precondition: User has selected withdraw option
- Main flow:
 - ▶ Include (Verify user)
 - ▶ Prompt user for amount to withdraw
 - ▶ Check available funds of user
 - ▶ Check available money of ATM
 - ▶ Remove amount from account
 - ▶ Give money
 - ▶ (print receipt)
 - ▶ Print current balance
- Exceptional flow
 - ▶ If not sufficient funds or money available, prompt user for lower amount

Soluzione

Use case: deposit

- Precondition: User has selected deposit option
- Main flow:
 - ▶ Include (Verify user)
 - ▶ Prompt user for amount of deposit
 - ▶ Open slot
 - ▶ Get check
 - ▶ (print receipt)
 - ▶ Print (balance + deposited amount)

Soluzione

Use case: check balance

- Precondition: User has selected balance option
- Main flow:
 - ▶ Include (Verify user)
 - ▶ Print balance

Soluzione

Use case: verify user

- Precondition: none
- Main flow:
 - ▶ User enters ID card
 - ▶ User enters PIN number
 - ▶ System checks validity of card and number
- Exceptional flow:
 - ▶ If combination is not valid, reject user

Soluzione

Use case: withdraw with receipt

- Precondition: User has selected withdraw option and print receipt option
- ...

Use case: deposit with receipt

- Precondition: User has selected deposit option and print receipt option item ...

Errori tipici sui diagrammi

- Diagrammi di flusso invece di casi d'uso: un caso d'uso è una sequenza di azioni, non una singola azione!
- Nome del caso d'uso che appare più volte nel diagramma
- Le frecce tra i casi d'uso non sono tratteggiate (- - - - ->) o etichettate «*extend*» o «*include*»
- «*extend*»: la freccia va dal caso che descrive l'evento alternativo al caso standard
- «*include*»: la freccia va dal caso chiamante al caso che descrive le azioni da includere

Errori tipici sugli scenari

- Assenza di precondizioni
- Mancata connessione alla rappresentazione grafica
- Nomi diversi per le stesse entità nelle rappresentazioni grafica e testuale
- Flusso eccezionale: mancanza di indicazioni nel flusso principale del punto in cui va controllata la condizione eccezionale

Riassumendo

- Analizzare il materiale contenente i requisiti
- Creare un glossario di progetto con parole chiave e una breve descrizione
- Capire chi sono gli attori
- Estrarre i casi d'uso più evidenti
- Cominciare ad organizzare i casi d'uso in un diagramma
- Modellare i casi d'uso come sequenze di eventi
- Raffinare progressivamente se necessario

Conclusioni

- Il diagramma UML dei casi d'uso è un tool per la modellazione del comportamento di un sistema.
- Descrive gli attori che interagiscono con il sistema, cosa fanno, e cosa ottengono dal sistema.
- A questo punto non interessa sapere come il sistema fornisca il comportamento richiesto.

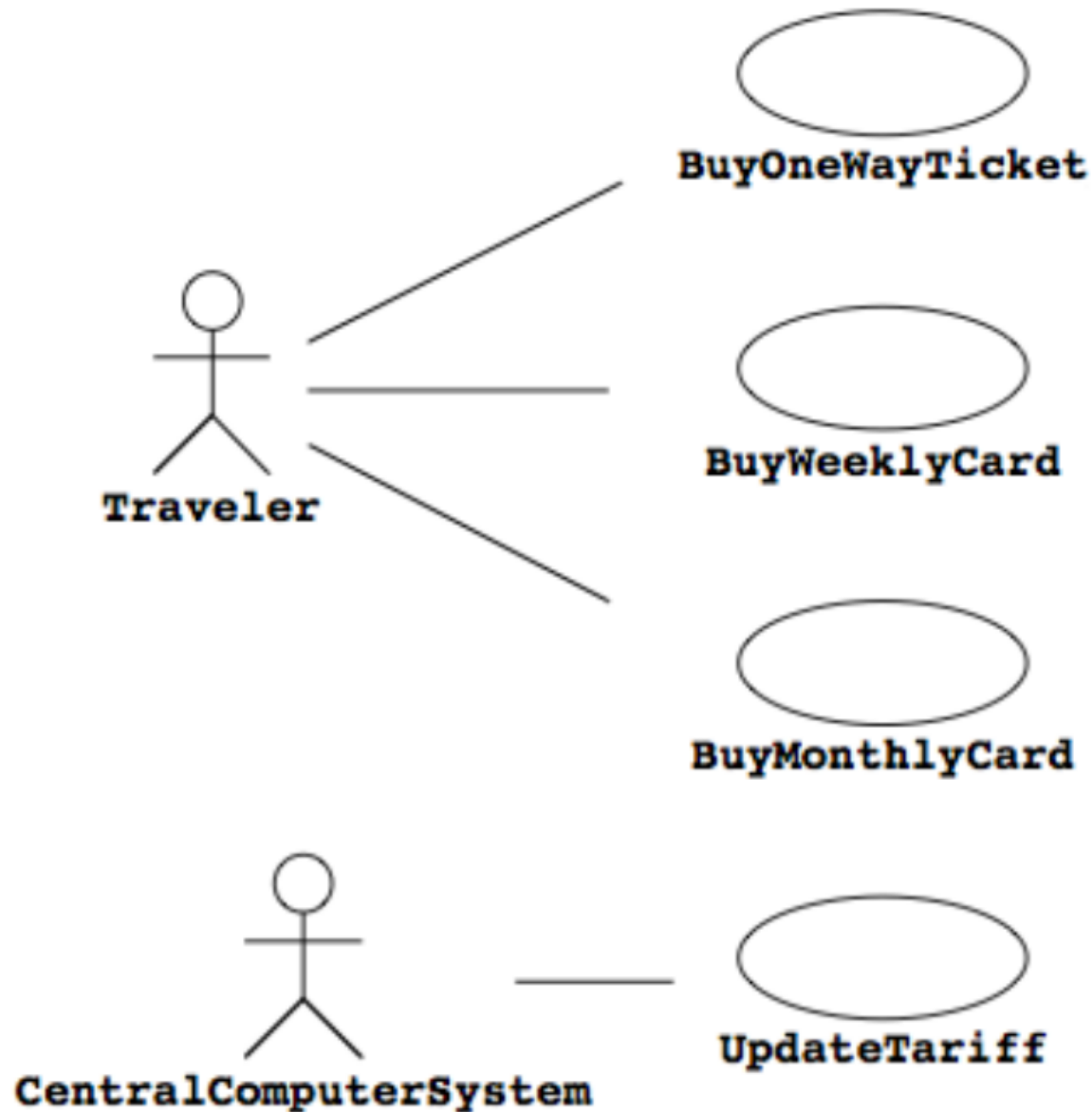
Esercizi con soluzioni

- “As always, there is never a 'correct' solution to any modelling problem. It's more that some models are more precise, and more informative, than others. You may find your solutions differ slightly from the ones below, but these solutions demonstrate good practice in model building.”
 - ▶ *dal corso “The Models Are the Code - Executable UML”, prof. Paul Krause, University of Surrey*

Esercizio biglietteria treno

- Disegnare un diagramma dei casi d'uso relativo ad una biglietteria elettronica (es. ferroviaria). Il sistema è usato dai viaggiatori, che comprano i biglietti, e da sistema centralizzato che aggiorna i costi dei biglietti.
- Gli utenti possono fare tre tipi di acquisti: biglietto di sola andata, abbonamento settimanale, abbonamento mensile.
- L'acquisto di un biglietto può fallire per quattro motivi: l'acquirente ha impiegato troppo tempo per completare l'acquisto, l'acquirente ha richiesto esplicitamente di non completare, la biglietteria ha esaurito il resto, la biglietteria ha esaurito la carta.

Attori e casi d'uso primari



Eccezioni ed estensioni

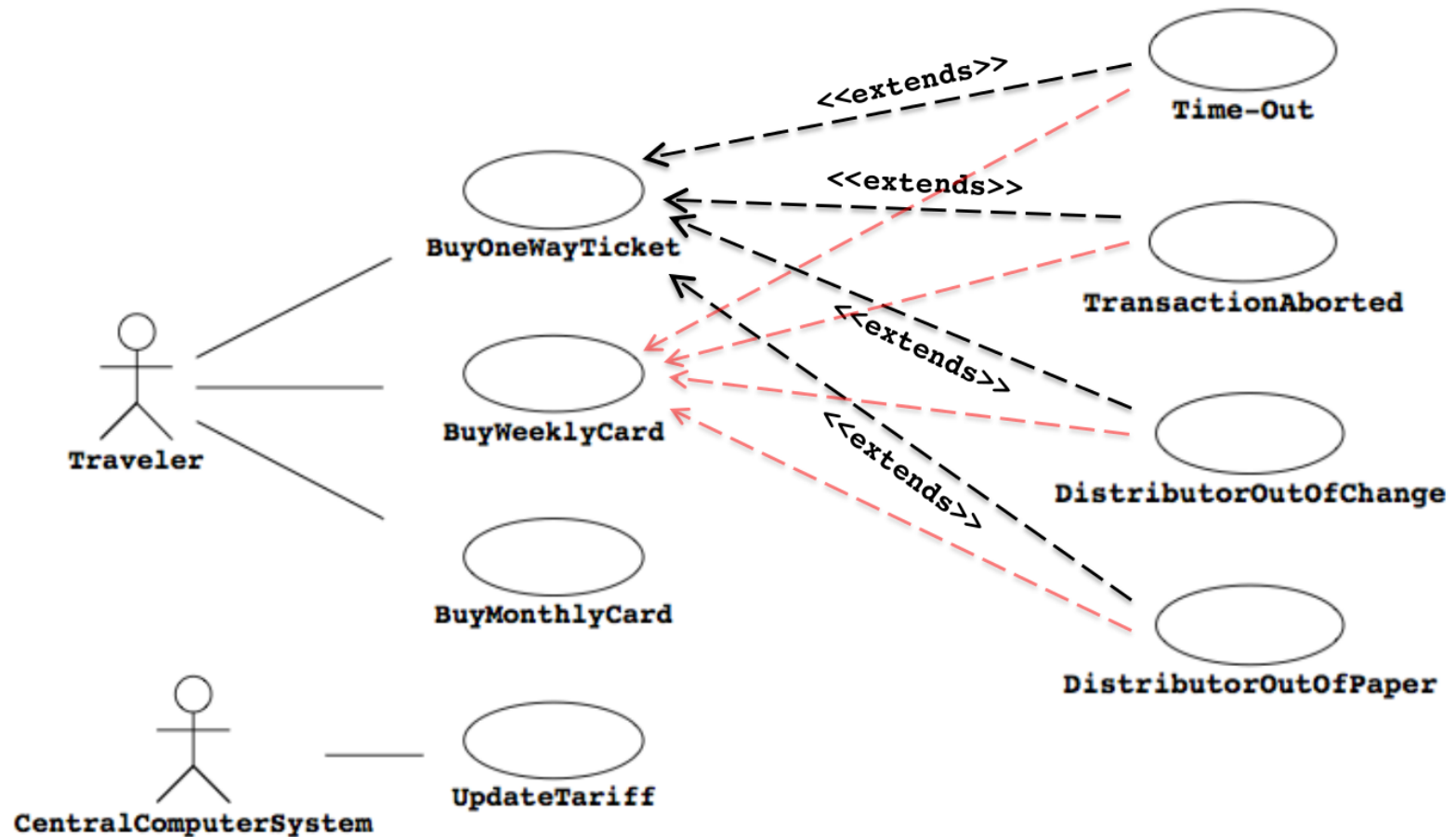
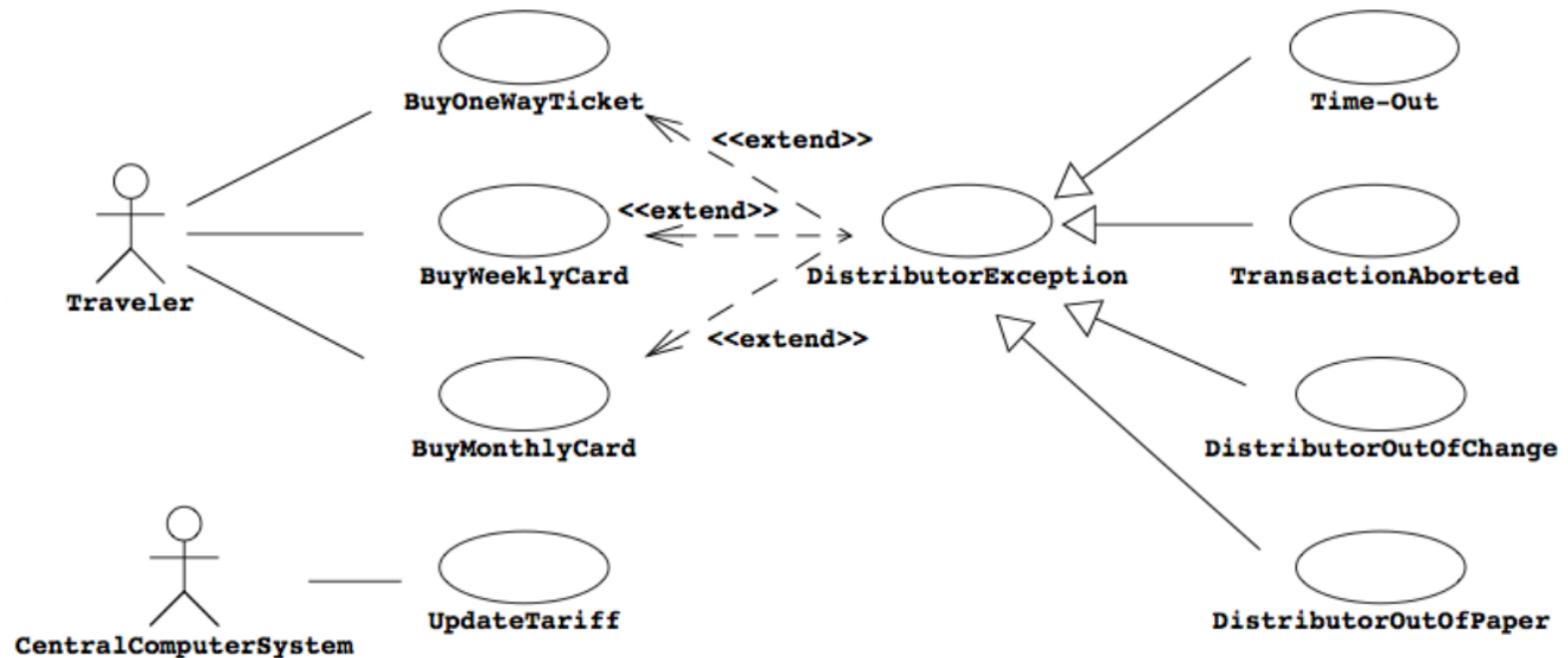


Diagramma completo



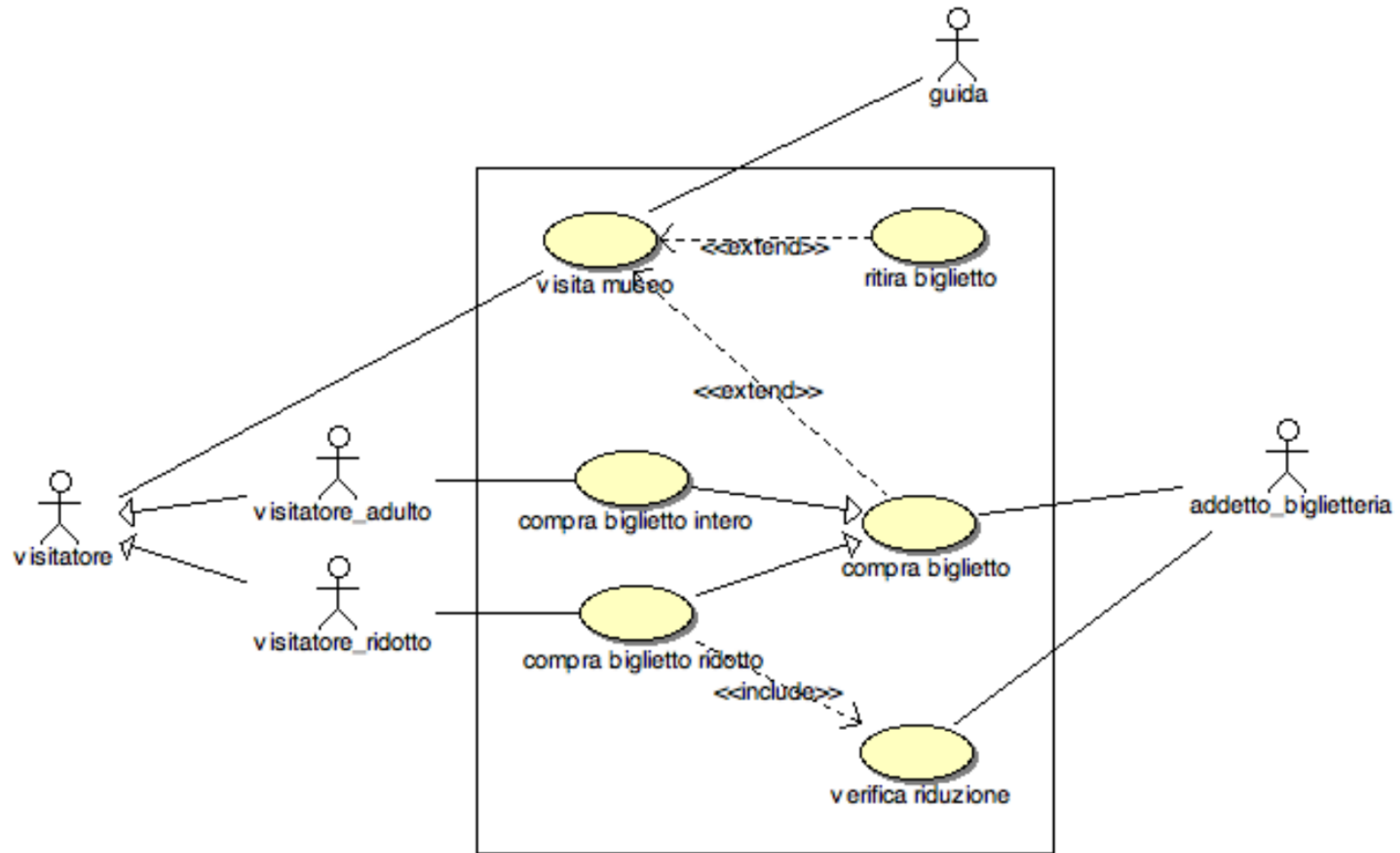
Esercizio museo

- Si consideri un sistema Museo. Gli utenti possono visitare il museo, comprando un biglietto venduto da un addetto alla biglietteria o usando biglietti acquistati precedentemente. La visite avvengono da soli oppure con una guida. Alcune categorie di visitatori hanno diritto ad un biglietto ridotto, previa dimostrazione dell'applicabilità della riduzione.
- Si rappresenti il sistema con un diagramma dei casi d'uso.

Esercizio museo: attori

- Si consideri un sistema Museo. Gli **utenti** possono visitare il museo, comprando un biglietto venduto da un **addetto alla biglietteria** o usando biglietti acquistati precedentemente. La visite avvengono da soli oppure con una **guida**. **Alcune categorie di visitatori** hanno diritto ad un biglietto ridotto, previa dimostrazione dell'applicabilità della riduzione.
- Si rappresenti il sistema con un diagramma dei casi d'uso.

Diagramma completo

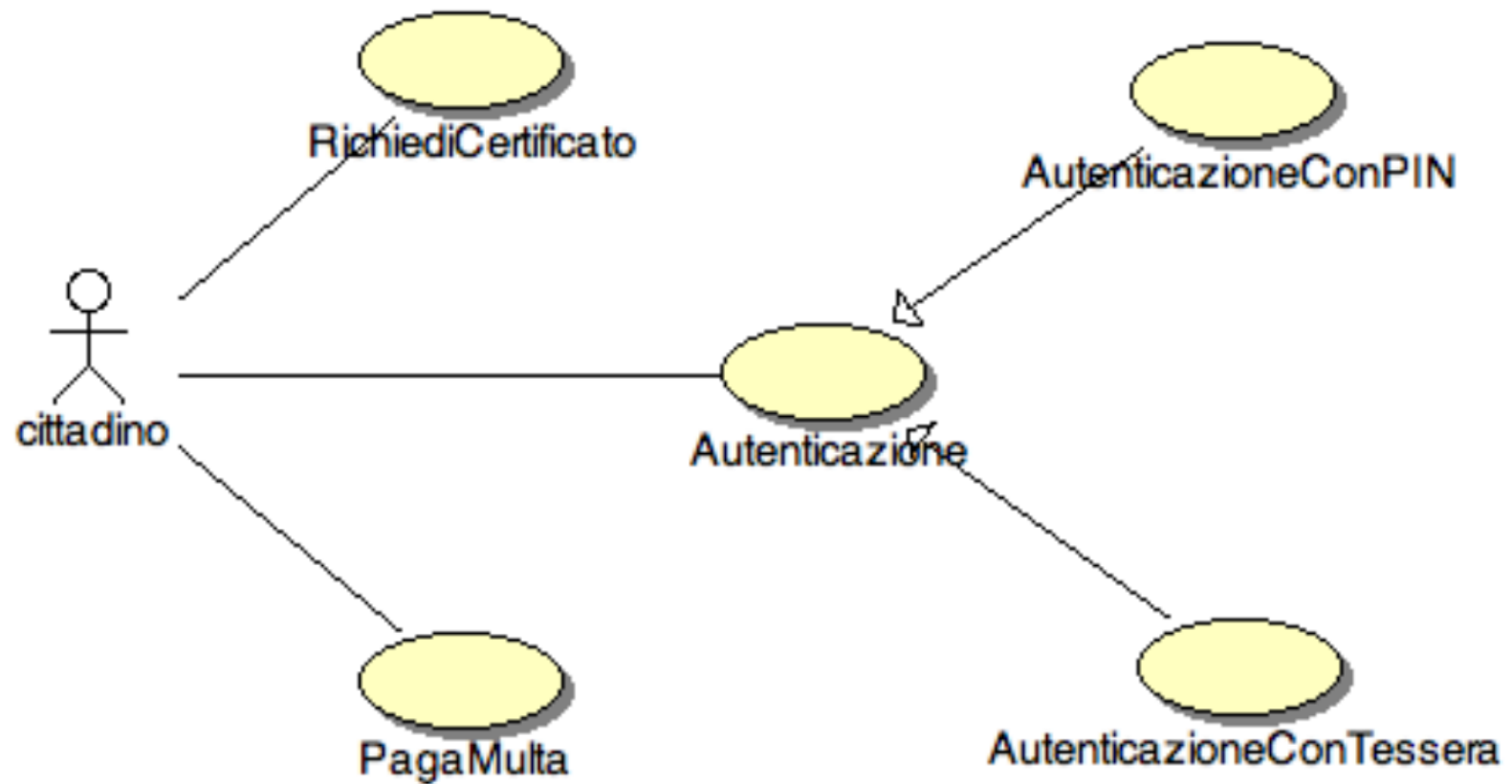


- Altre soluzioni?
- Differenziare le visite e/o gli acquisti?

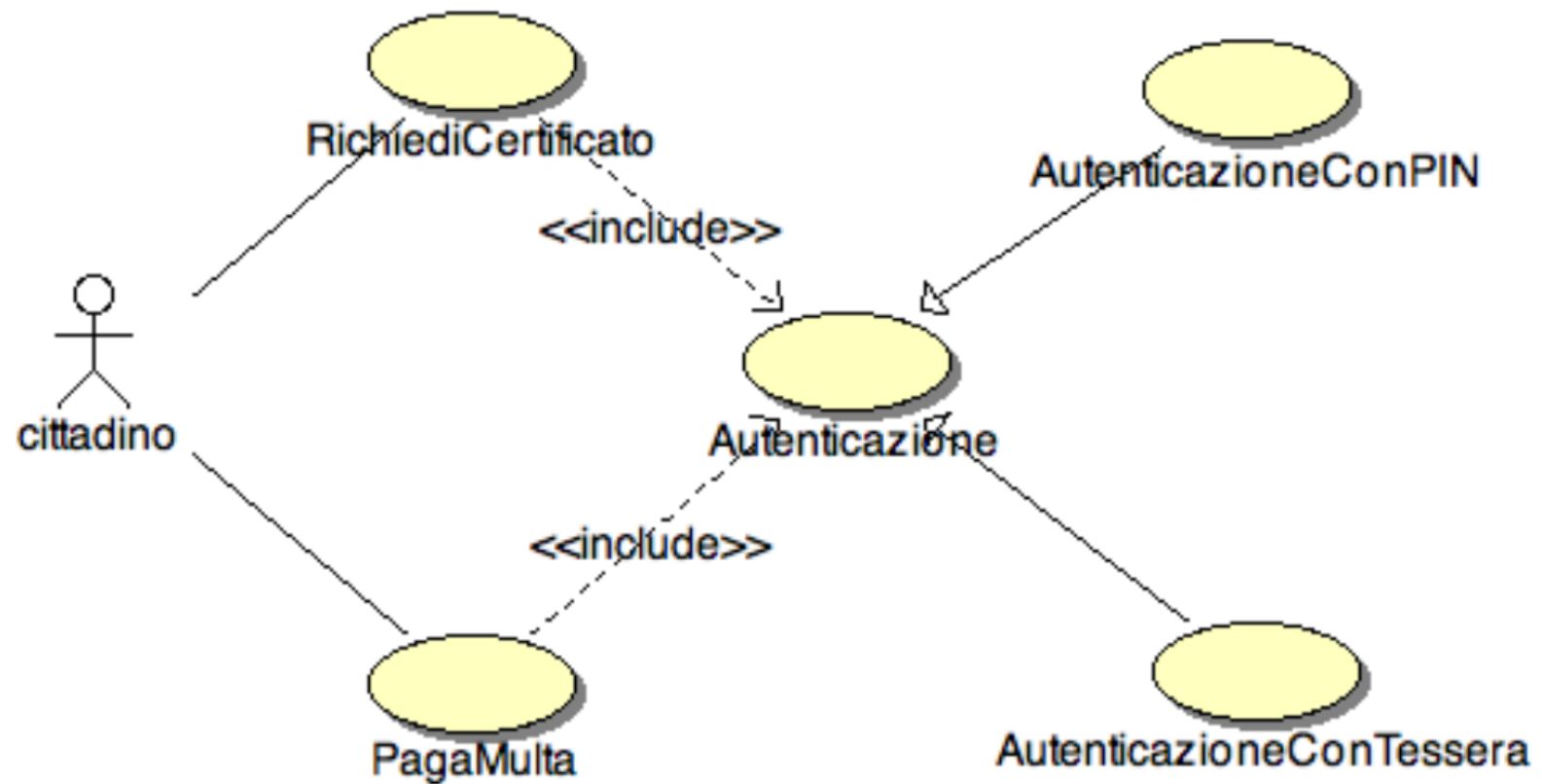
Esercizio sportello del cittadino

- Si consideri un sistema di sportello automatico, da cui i cittadini possono ritirare certificati o pagare multe, previa autenticazione tramite tessera magnetica o inserimento di un PIN personale.
- Si rappresenti il sistema con un diagramma dei casi d'uso.

Esercizio: cosa non va?



Esercizio: diagramma



Esercizio negozio online

- Si consideri un negozio che rende disponibile un catalogo liberamente consultabile on-line. Gli utenti registrati possono inviare un ordine di acquisto (comunicando i dati di pagamento), che viene memorizzato nel sistema e trasferito al reparto ordini che lo evade.
- Si rappresenti il sistema con un diagramma dei casi d'uso.

Esercizio: Estrarre i requisiti

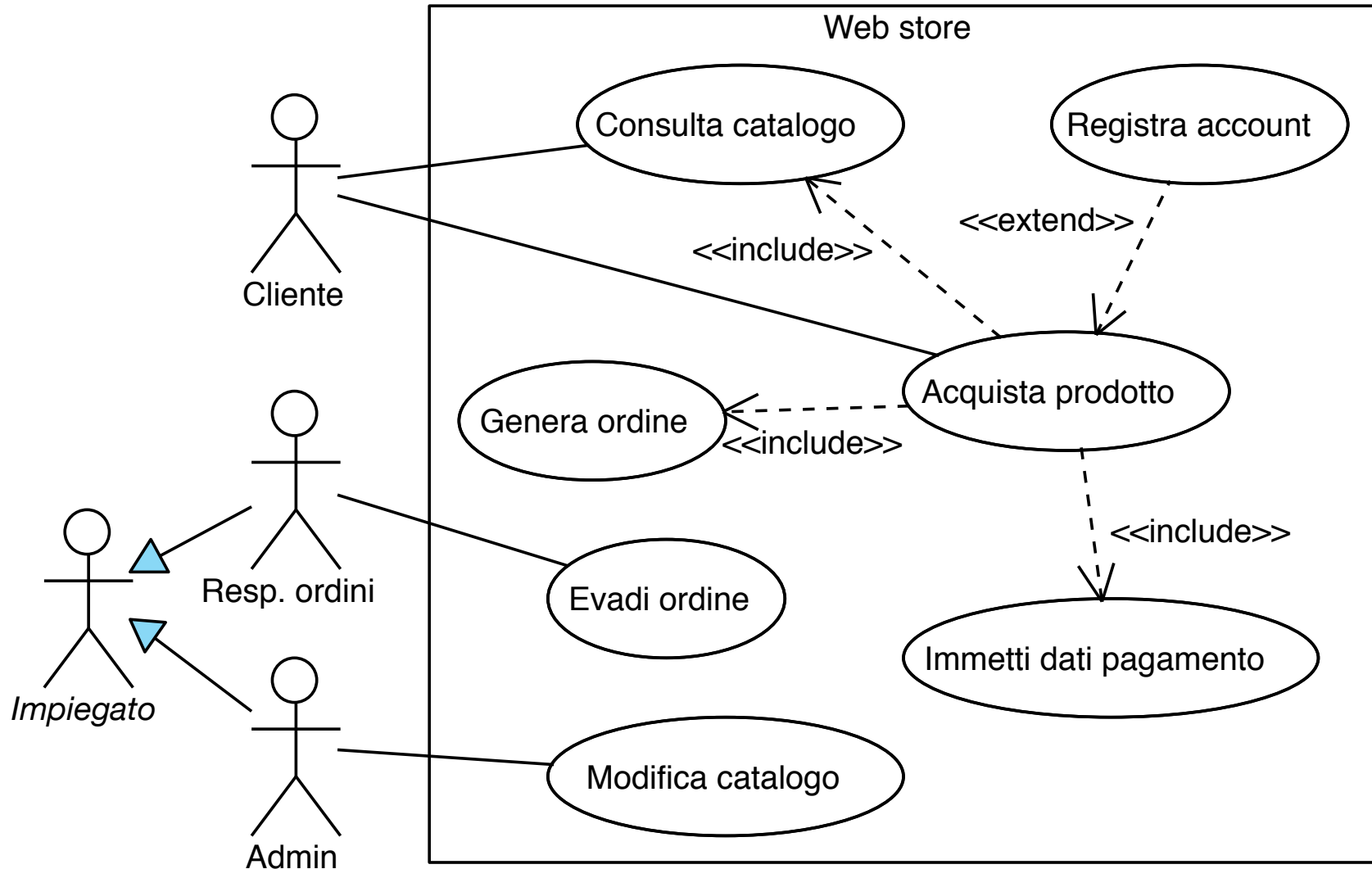
Chi interagisce con il sistema (**attori**)?

- Clienti
- Amministratori del negozio online
- Reparto ordini

Cosa fanno (**casi d'uso**)?

- Il cliente si registra, consulta il catalogo ed effettua acquisti
- L'amministratore organizza il catalogo, che è diviso in categorie
- Il reparto ordini riceve ordini da evadere
- Il cliente sceglie il tipo di pagamento

Esercizio: diagramma



Esercizio immetti pagamento

- Si consideri l'esercizio precedente relativo al catalogo on-line. Scrivere la specifica del caso d'uso 'Immetti pagamento'.

Esercizio: soluzione

ImmettiPagamento	
ID	CU1
Attori	Utente
Precondizioni	Il Cliente ha selezionato i prodotti da acquistare
Sequenza	
Postcondizioni	

Esercizio: cosa non va?

- Incomincia quando si seleziona la funzione 'immetti pagamento'
- Vengono inseriti i dati del cliente
- Il sistema verifica i dati del cliente

Esercizio: cosa non va?

- ~~Incomincia quando si seleziona la funzione 'immetti pagamento'~~
- Il caso d'uso inizia quando il cliente seleziona la funzione 'immetti pagamento'
- ~~Vengono inseriti i dati del cliente~~
- Il cliente inserisce nel form il suo nome e il numero di carta di credito
- Il sistema verifica i dati del cliente

Esercizio: soluzione

ImmettiPagamento	
ID	CU1
Attori	Utente
Precondizioni	Il Cliente ha selezionato i prodotti da acquistare
Sequenza	<ul style="list-style-type: none">• Il caso d'uso inizia quando il cliente seleziona la funzione `immetti pagamento`• Il cliente inserisce nel form il suo nome e il numero di carta di credito• Il cliente invia i dati• Il sistema verifica i dati del cliente• Il cliente conferma il pagamento
Postcondizioni	Il sistema riceve i dati dal pagamento e può generare l'ordine.

Testi esercizi

Esercizio gestione biblioteca

- Viene richiesto un sistema che permetta al bibliotecario e a un utente di effettuare ricerche di libri. Il bibliotecario deve poter effettuare il prestito e gestire la restituzione del libro. Un utente deve restituire il libro entro una certa data. Se il prestito risulta scaduto per la prima volta il sistema emette un avviso, se è la seconda volta il bibliotecario registra e stampa una multa. L'utente a questo punto può decidere se pagare la multa subito oppure no. Il sistema deve permettere la registrazione del pagamento.
- Disegnare il diagramma dei casi d'uso e descrivere le sequenze relative.

Esercizio gestione del personale

- Viene richiesto un sistema che permetta la gestione del personale di un'azienda. Per accedere alle operazioni bisogna autenticarsi. Le operazioni possibili saranno la modifica dei dati dell'impiegato, la semplice visualizzazione dei suoi dati, e la cancellazione dei dati.
- Disegnare il diagramma dei casi d'uso e descrivere le sequenze relative.

Esercizio Ricerca di un prodotto

- Una libreria ha la necessità di un sistema che le permetta di far effettuare ricerche al suo personale. All'interno della libreria vengono venduti anche dvd video, e cd musicali.
- Disegnare il diagramma dei casi d'uso e descrivere le sequenze relative.