

11. Analisi dei requisiti (IV)

Prof. Paola Barra
a.a. 2022/2023

Associazioni nei casi d'uso

Associazione nei casi d'uso

- Un modello dei casi d'uso consiste di casi d'uso ed associazioni di casi d'uso
- Un'associazione è una relazione tra casi d'uso
- Le associazioni più importanti sono: *Communication, Include, Extend, Generalization*

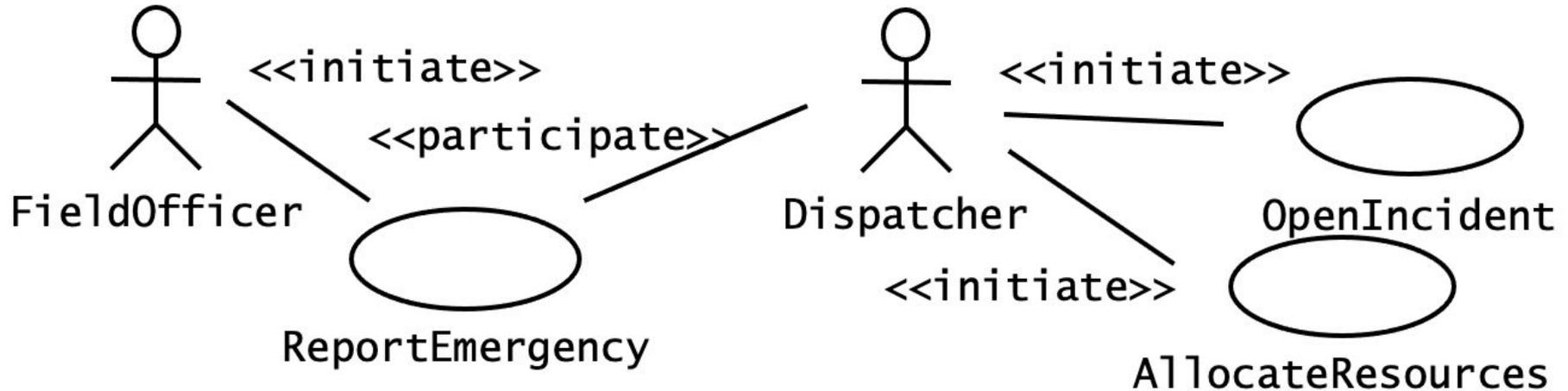
Identificazione delle relazioni tra attori e casi d'uso

- Anche sistemi di medie dimensioni hanno numerosi casi d'uso
- Le relazioni tra gli attori ed i casi d'uso permettono a sviluppatori ed utenti di ridurre la complessità del modello ed aumentarne la comprensione
 - Si usano **relazioni di comunicazione** tra attori e casi d'uso per descrivere il sistema in livelli di funzionalità
 - Le **relazioni di estensione** sono usate per separare flussi di eventi eccezionali da flussi di eventi comuni
 - Le **relazioni di inclusione** sono usate per ridurre la ridondanza tra casi d'uso
 - Le **generalizzazioni** specializzano casi d'uso astratti

Relazioni di comunicazione

- Rappresentano il flusso di informazioni durante il caso d'uso
 - L'attore che **inizia** il caso d'uso dovrebbe essere distinto da altri attori con cui il caso d'uso comunica
 - Specificando quale attore può invocare uno specifico caso d'uso, specifichiamo implicitamente quale attore non può invocare il caso d'uso
 - Similmente, specificando quali attori comunicano con uno specifico caso d'uso, specifichiamo quali attori accedono informazioni specifiche e quali non possono
 - In definitiva, documentando l'iniziazione e le relazioni di comunicazioni tra attori e casi d'uso, specifichiamo grosso modo il controllo di accesso per il sistema
- Le relazioni tra attori e casi d'uso sono identificate quando sono identificati i casi d'uso

Esempio di relazioni di comunicazione tra attori e casi d'uso in FRIEND



Associazione Extend

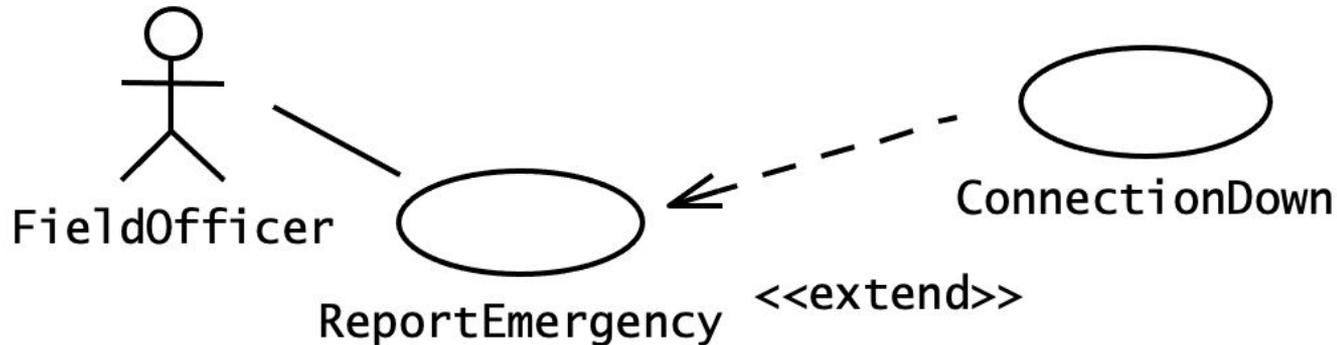
- Problema
 - La funzionalità nel problema originale ha la necessità di essere estesa
- Soluzione
 - Un'associazione *extend* dal caso d'uso B al caso d'uso A indica che il caso d'uso B è un'estensione del caso d'uso A

Relazione EXTEND tra casi d'uso

- Un caso d'uso estende un altro caso d'uso se il caso d'uso esteso può inglobare il comportamento dell'estensione sotto certe condizioni
- In FRIEND, supponiamo che la connessione tra le stazioni di *FieldOfficer* e *Dispatcher* sia interrotta mentre il *FieldOfficer* sta compilando il form (ad esempio, l'auto del *FieldOfficer* entra in un tunnel)
 - La stazione del *FieldOfficer* deve notificare il *FieldOfficer* che il suo form non è stato consegnato e quali misure dovrebbe intraprendere
 - Il caso d'uso *ConnectionDown* viene rappresentato come un'estensione di *ReportEmergency*
 - Le condizioni sotto cui il caso d'uso *ConnectionDown* è iniziato sono descritte in *ConnectionDown* anziché in *ReportEmergency*

Utilizzo della relazione EXTEND

Osservazione: nelle associazioni *extend*, il caso d'uso base può essere eseguito senza la sua estensione

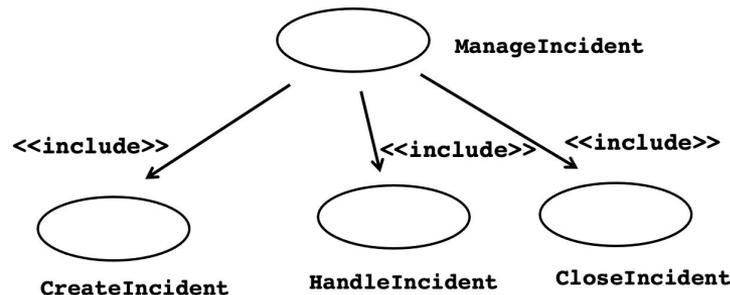


Relazione EXTEND tra casi d'uso

- Separare flussi di eventi eccezionali ed opzionali dal caso d'uso base comporta due vantaggi
 - Il caso d'uso base è reso più breve e più semplice da capire
 - Il caso d'uso comune è distinto dal caso eccezionale, ciò consente agli sviluppatori di trattare ogni tipo di funzionalità in modo differente
 - Ottimizzare il caso d'uso comune rispetto al tempo di risposta
 - Ottimizzare il caso eccezionale rispetto alla robustezza
- Ambo i casi d'uso esteso ed estensione sono casi d'uso completi
 - Devono avere condizioni di entrata ed uscita ed essere comprensibili all'utente in modo indipendente

Associazione INCLUDE: decomposizione funzionale

- Problema
 - Una funzione nella definizione del problema originale è troppo complessa per risolverla immediatamente
- Soluzione
 - Descrivere la funzione come un'aggregazione di un insieme di funzioni più semplici. Il caso d'uso associato è decomposto in casi d'uso più piccoli



INCLUDE: riuso di funzionalità esistenti

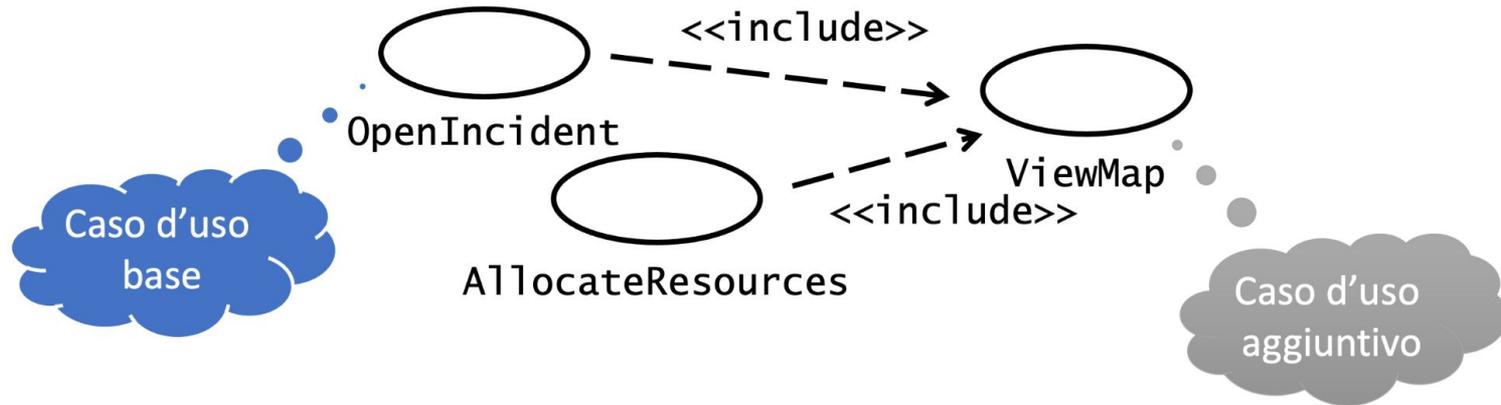
- Problema
 - Ci sono già funzioni esistenti. Come possiamo riusarle?
- Soluzione
 - L'associazione *include* da un caso d'uso A ad un caso d'uso B indica che un'istanza del caso d'uso A esegue tutti i comportamenti descritti nel caso d'uso B (“A delega a B”)

Relazione INCLUDE tra casi d'uso

- In questo modo la ridondanza tra casi d'uso può essere esplicitata usando la relazione di inclusione
 - Il *Dispatcher* deve consultare la mappa della città quando inizia la gestione di un incidente (ad esempio, per valutare quali zone sono a rischio durante un incendio) e quando deve allocare le risorse sul luogo (per verificare quali sono le risorse più vicine)
 - Il caso d'uso *ViewMap* descrive il flusso degli eventi richiesto quando si consulta la mappa
 - Usato sia da *OpenIncident* che da *AllocateResources*

Relazione INCLUDE tra casi d'uso

Osservazione: il caso base non può esistere da solo. E' sempre invocato insieme al caso d'uso aggiuntivo



Relazione INCLUDE tra casi d'uso

- Esplicitare comportamenti condivisi dai casi d'uso ha molti benefici
 - Descrizioni più brevi e meno ridondanze
- Il comportamento dovrebbe essere esplicitato in un caso d'uso separato solo se è condiviso tra due o più casi d'uso
- Un'eccessiva frammentazione della specifica dei requisiti attraverso un elevato numero di casi d'uso rende confusa la specifica ai clienti ed agli utenti

Relazioni EXTEND vs relazioni INCLUDE

- Entrambe le relazioni sono simili
 - Inizialmente potrebbe non essere chiaro allo sviluppatore quando usarle
- La principale distinzione tra le due relazioni sta nella direzione della relazione
 - Per le relazioni *include*, l'evento che innesca il caso d'uso target (quello incluso) è descritto nel flusso di eventi del caso d'uso sorgente
 - Per le relazioni *extend*, l'evento che innesca il caso d'uso sorgente (quello che estende) è descritto nel caso d'uso sorgente come condizione
 - In altre parole, per le relazioni *include*, ogni caso d'uso che include deve specificare dove il caso d'uso incluso dovrebbe essere invocato
 - Per le relazioni *extend*, solo il caso d'uso che estende specifica quali casi d'uso sono estesi

ReportEmergency (relazione include)

1. ...
2. ...
3. Il *FieldOfficer* completa il form selezionando il livello, il tipo, l'ubicazione dell'emergenza e una breve descrizione della situazione. Il *FieldOfficer* descrive anche le risposte possibili alla situazione di emergenza. Una volta che il form è completo, il *FieldOfficer* sottomette il form, al cui punto, è notificato il *Dispatcher*. **Se la connessione con il Dispatcher è interrotta, è usato il caso d'uso ConnectionDown.**
4. Se la connessione è ancora valida, il *Dispatcher* rivede le informazioni sottomesse e crea un *Incident* nel db invocando il caso d'uso *OpenIncident*. Il *Dispatcher* seleziona una risposta e accetta il rapporto di emergenza. **Se la connessione è interrotta, è usato il caso d'uso ConnectionDown.**
5. ...

ReportEmergency (relazione extend)

1. ...
2. ...
3. Il *FieldOfficer* completa il form selezionando il livello, il tipo, l'ubicazione dell'emergenza e una breve descrizione della situazione. Il *FieldOfficer* descrive anche le risposte possibili alla situazione di emergenza. Una volta che il form è completo, il *FieldOfficer* sottomette il form, al cui punto, è notificato il *Dispatcher*.
4. Il *Dispatcher* rivede le informazioni sottomesse e crea un *Incident* nel db invocando il caso d'uso *OpenIncident*. Il *Dispatcher* seleziona una risposta e accetta il rapporto di emergenza.
5. ...

ConnectionDown (relazione include)

1. Il FieldOfficer e il Dispatcher sono notificati dell'interruzione della connessione. Sono avvisati delle possibili ragioni per cui si è verificato l'evento (ad esempio, "la stazione del FieldOfficer è in un tunnel?")
2. La situazione vien trascritta dal sistema e recuperata quando la connessione è ristabilita.
3. Il FieldOfficer e il Dispatcher entrano in contatto in altri modi ed il Dispatcher inizia ReportEmergency dalla stazione del Dispatcher.

ConnectionDown (relazione extend)

Il caso d'uso ConnectionDown estende qualsiasi caso d'uso in cui la comunicazione tra il FieldOfficer e il Dispatcher può essere persa.

1. Il FieldOfficer e il Dispatcher sono notificati dell'interruzione della connessione. Sono avvisati delle possibili ragioni per cui si è verificato l'evento (ad esempio, "la stazione del FieldOfficer è in un tunnel?")
2. La situazione vien trascritta dal sistema e recuperata quando la connessione è ristabilita.
3. Il FieldOfficer e il Dispatcher entrano in contatto in altri modi ed il Dispatcher inizia ReportEmergency dalla stazione del Dispatcher.

Relazioni EXTEND vs relazioni INCLUDE

- Nella colonna di sinistra (relazione include), dobbiamo inserire il testo in due punti nel flusso di eventi dove il caso d'uso **ConnectionDown** può essere invocato
 - Inoltre, se sono descritte situazioni eccezionali aggiuntive (un funzione help sulla stazione del **FieldOfficer**), il caso d'uso **ReportEmergency** dovrà essere modificato e sarà ingombrato dalle condizioni

Relazioni EXTEND vs relazioni INCLUDE

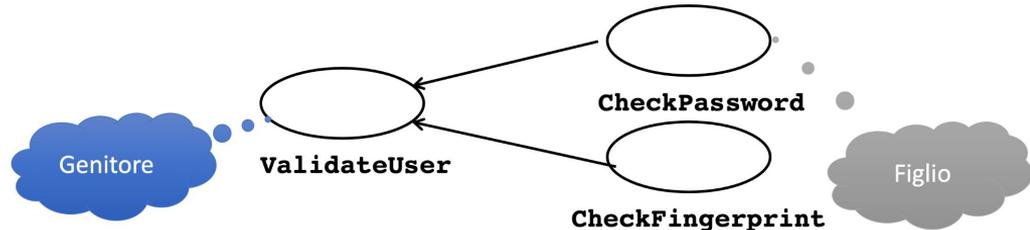
- Nella colonna di destra, dobbiamo descrivere solo le condizioni sotto cui il caso d'uso eccezionale è invocato, il che può includere numerosi casi d'uso
 - Inoltre, situazioni eccezionali aggiuntive possono essere aggiunte senza modificare il caso d'uso base (***EmergencyReport***)
- L'abilità di estendere il sistema senza modificare parti esistenti è critica, poiché consente di lasciare inalterato il comportamento originale
- La distinzione tra *include* e *extend* è una questione di documentazione
 - Usare il tipo giusto di relazione riduce le dipendenze tra i casi d'uso, la ridondanza, e abbassa la probabilità di introdurre errori quando cambiano i requisiti

Euristiche per le relazioni **EXTEND** e **INCLUDE**

- Usare le relazioni **extend** per comportamenti eccezionali, opzionali o che si verificano di rado
 - Un esempio di comportamento raro è il guasto di una risorsa
 - Un esempio di comportamento opzionale è la notifica di risorse nelle vicinanze che rispondono ad un incidente che non le riguarda
- Usare le relazioni **include** per un comportamento condiviso tra due o più casi d'uso
- Usare con discrezione le due euristiche precedenti e non sovra strutturare il modello del caso d'uso. Casi d'uso un po' più lunghi (due pagine) sono più semplici da capire e rivedere rispetto a molti casi d'uso brevi (lunghi dieci righe, ad esempio)

Associazione di generalizzazione tra casi d'uso

- Problema
 - C'è un comportamento comune ma più specifico tra casi d'uso e vogliamo esplicitarlo
- Soluzione
 - L'associazione di generalizzazione tra casi d'uso esplicita comportamenti comuni. I casi d'uso figli ereditano il comportamento ed il significato del genitore ed aggiungono altri comportamenti
- Esempio
 - Consideriamo il caso d'uso **ValidateUser**, responsabile della verifica dell'identità di un utente. Il cliente potrebbe richiedere due realizzazioni: **CheckPassword** e **CheckFingerprint**



Identificazione degli oggetti di analisi iniziali

- Uno dei primi ostacoli che sviluppatori ed utenti incontrano quando cominciano a collaborare è la terminologia differente
 - Anche se gli sviluppatori imparano la terminologia degli utenti, il problema si pone ancora quando si aggiungono nuovi sviluppatori al progetto
 - Le incomprensioni si verificano per l'uso di stessi termini usati in contesti differenti con significati diversi
- Per stabilire una terminologia chiara, gli sviluppatori identificano gli oggetti partecipanti per ogni caso d'uso
 - Devono identificare il nome, descriverli in modo non ambiguo e raccogliarli in un glossario
 - La costruzione del glossario costituisce il primo passo verso l'analisi

Glossario

- Il glossario è incluso nella specifica dei requisiti e successivamente nei manuali utente
- Gli sviluppatori mantengono il glossario aggiornato durante l'evoluzione della specifica dei requisiti
- I benefici del glossario sono diversi
 - I nuovi sviluppatori hanno a disposizione un insieme consistente di definizioni
 - E' usato un termine singolo per ogni concetto
 - Ogni termine ha un preciso e chiaro significato ufficiale

Identificazione degli oggetti di analisi iniziale

- L'identificazione degli oggetti partecipanti da luogo al modello ad oggetti di analisi iniziale
 - Questo passo, durante la scoperta dei requisiti, costituisce solo un primo passo verso il modello degli oggetti di analisi completo
 - Il modello di analisi completo solitamente non è usato come mezzo di comunicazione per sviluppatori ed utenti
 - Gli utenti il più delle volte non hanno familiarità con concetti orientati agli oggetti
 - Tuttavia, la descrizione degli oggetti ed i loro attributi sono visibili agli utenti e revisionati

Euristiche per l'identificazione degli oggetti di analisi iniziale

- Termini che gli sviluppatori o gli utenti devono chiarire per capire i casi d'uso
- Sostantivi ricorrenti nei casi d'uso (ad esempio, **Incident**)
- Entità del mondo reale di cui il sistema deve tenere traccia (ad esempio, **FieldOfficer** e **Resource**)
- Processi del mondo reale di cui il sistema deve tenere traccia (**EmergencyOperationPlan**)
- Casi d'uso (**ReportEmergency**)
- Sorgenti di dati (**Printer**)
- Artefatti con cui l'utente interagisce (**Station**)
- Usare sempre i termini del dominio dell'applicazione

Identificazione degli oggetti

- Durante la scoperta dei requisiti, sono generati gli oggetti partecipanti per ciascun caso d'uso
 - Se due casi d'uso si riferiscono allo stesso concetto, l'oggetto corrispondente dovrebbe essere lo stesso
 - Se due oggetti condividono lo stesso nome e non corrispondono allo stesso concetto, uno o entrambi i concetti sono rinominati per enfatizzarne la differenza
 - Si eliminano così le ambiguità nella terminologia impiegata

Oggetti partecipanti al caso d'uso *ReportEmergency*

Dispatcher	Ufficiale di Polizia che gestisce gli Incident. Un Dispatcher apre, documenta, e chiude gli incidenti in risposta ad un EmergencyReport e altre comunicazioni con i FieldOfficer. I Dispatcher sono identificati con numeri di badge.
EmergencyReport	Rapporto iniziale su un Incident da un FieldOfficer ad un Dispatcher. Un EmergencyReport solitamente innesca la creazione di un Incident dal Dispatcher. Un EmergencyReport è composto da un livello di emergenza, un tipo (fuoco, incidente stradale, altro), una località e una descrizione.
FieldOfficer	Ufficiale di Polizia o dei Vigili del fuoco in servizio. Un FieldOfficer può essere allocato al più ad un incidente alla volta. I FieldOfficer sono identificati dai numeri di badge.
Incident	Situazione che richiede l'attenzione di un FieldOfficer. Il rapporto su un Incident può essere inserito nel sistema da un FieldOfficer o chiunque altro esterno al sistema. Un Incident è composto da una descrizione, una risposta, uno stato (aperto, chiuso, documentato), una località, e un numero di FieldOfficer.

Euristiche per il controllo incrociato di casi d'uso ed oggetti partecipanti

- Quali casi d'uso creano questo oggetto (cioè, durante quali casi d'uso i valori degli attributi di un oggetto sono immessi nel sistema)?
- Quali attori possono accedere a tali informazioni?
- Quali casi d'uso modificano e distruggono questo oggetto (cioè, quali casi d'uso editano o rimuovono questa informazione dal sistema)?
- Quale attore può iniziare questi casi d'uso?
- E' necessario questo oggetto (cioè, c'è almeno un caso d'uso che dipende da questa informazione)?

Identificazione dei requisiti non funzionali

- I requisiti non funzionali possono avere un impatto importante sul lavoro dell'utente in modo inaspettato
- Per scoprire in modo accurato tutti i requisiti non funzionali essenziali, ambo i clienti e sviluppatori devono collaborare in modo da identificare quali attributi del sistema (minimali) difficili da realizzare sono critici per il lavoro dell'utente
- L'insieme risultante di requisiti non funzionali tipicamente include requisiti in conflitto tra loro

Euristiche per i requisiti non funzionali

- Ci sono pochi metodi sistematici per scoprire i requisiti non funzionali
- Gli analisti usano una tassonomia (schema FURPS+) dei requisiti non funzionali per generare una lista di controllo di domande per aiutare clienti e sviluppatori a focalizzarsi sugli aspetti non funzionali del sistema
- Poiché gli attori in questa fase sono già stati identificati, la lista di controllo può essere organizzata per ruolo e distribuita agli utenti rappresentativi

Esercitazioni - casi d'uso

**Analisi dei requisiti
e modelli e astrazioni**

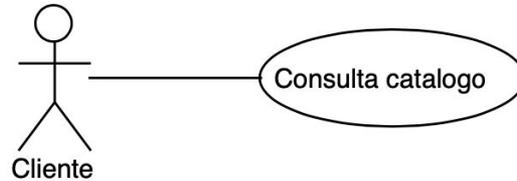
Tools UML

- ArgoUML, <http://argouml.tigris.org/>
- Eclipse MDT UML2, <http://www.eclipse.org/uml2/>
- Omondo EclipseUML, <http://www.omondo.com/>
- boUML, <http://bouml.free.fr/>
- Umbrello UML, <http://uml.sourceforge.net/>

Il diagramma dei casi d'uso

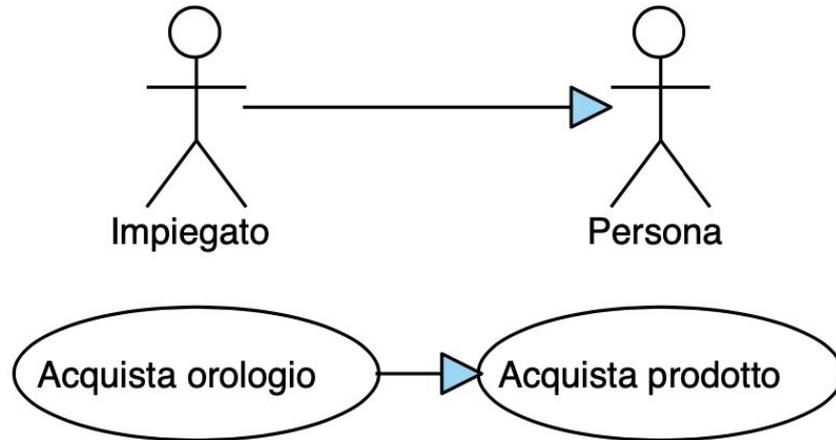
- Si tratta di un diagramma che esprime un comportamento, desiderato o offerto.
- Individua:
 - ▶ **chi** o che cosa ha a che fare con il sistema (**attore**)
 - ▶ **che cosa** l'attore può fare (**caso d'uso**).
- Modella i **requisiti funzionali** di un sistema.
 - ▶ I requisiti funzionali specificano cosa deve essere fatto.
 - ▶ Sono indipendenti dalla tecnologia, dall'architettura, dalla piattaforma, dal linguaggio di programmazione.
- Sono esclusi i requisiti **non-funzionali**, che specificano **vincoli aggiuntivi** (performance, scalabilità, ecc.)
- Si individuano prima gli attori e poi i casi d'uso!

Un po' di notazione UML



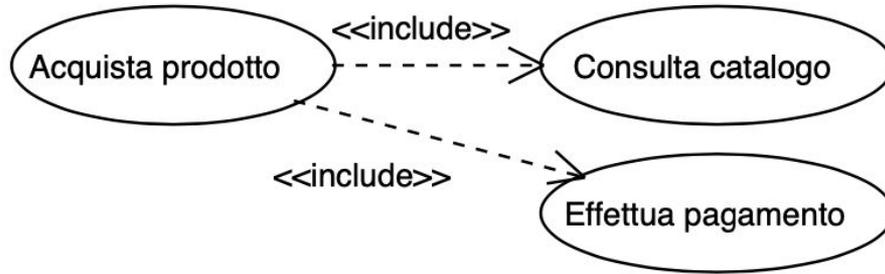
- **Caso d'uso:** specifica di una sequenza di azioni, incluse eventuali sequenze alternative e/o di errore che un sistema (o sottosistema) può eseguire interagendo con attori esterni.
 - ▶ Il nome (etichetta) dovrebbe essere basato su un verbo o su un sostantivo che esprime un avvenimento.
- **Attore:** un ruolo assunto da un utente o altra entità che interagisce col sistema nell'ambito di un caso d'uso.
 - ▶ Non è necessariamente umano: oggetto fisico, agente software, condizioni ambientali, etc.
- **Associazione:** collega gli attori ai casi d'uso.

Generalizzazione



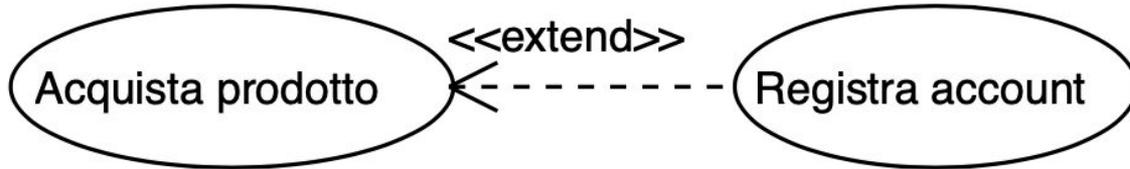
- Collega un attore o caso d'uso ad un altro più generale.
- Il figlio può sostituire il genitore dovunque questi appaia.

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.
- Usato per **riutilizzare parti comuni** a più casi d'uso.

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?

Esercizio 1: 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 1: negozio online - estrazione requisiti

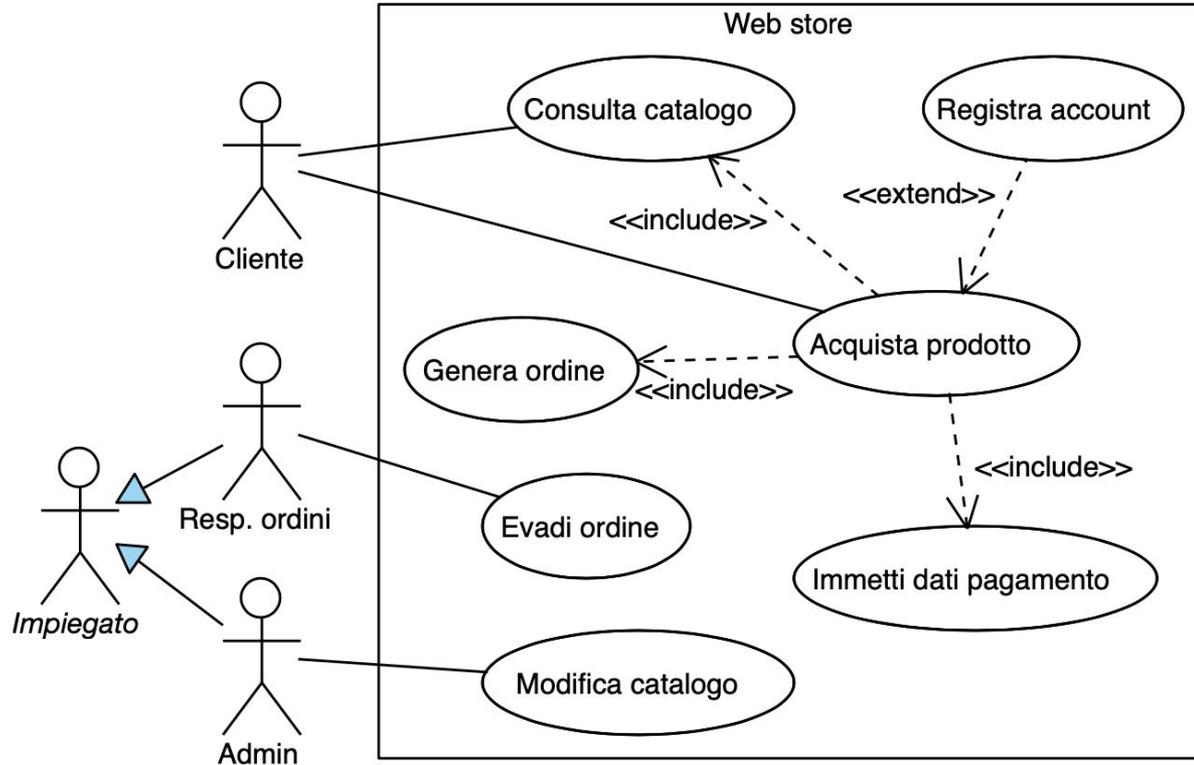
Chi interagisce con il sistema (**attori**)?

- Clienti
- Amministratori del negozio online
- Reparto ordini

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

- Il cliente si registra, consulta il catalogo ed effettua acquisti
- Il cliente sceglie il tipo di pagamento
- L'amministratore organizza il catalogo
- Il reparto ordini evade gli ordini

Esercizio 1: negozio online - soluzione



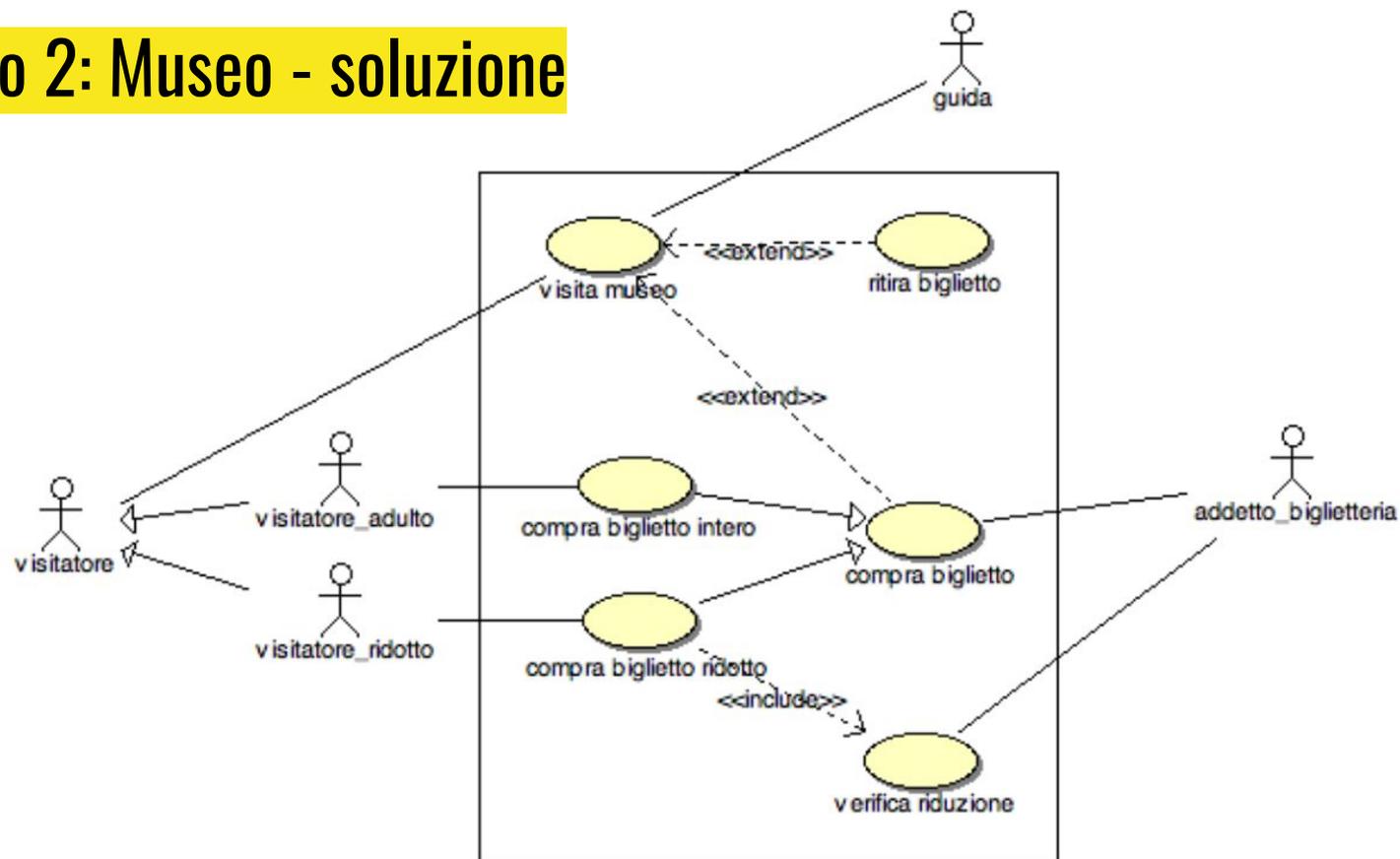
Esercizio 2: Museo

- 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 2: 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 2: Museo - soluzione



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

Esercizio 3 : negozio online → immetti pagamento

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

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

Esercizio 3 - sequenza di eventi

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

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.

Esercizio 4 : aggiorna carrello

- Si descriva la specifica del caso dal caso d'uso 'aggiorna carrello' di un negozio on-line.
- Dopo aver selezionato un articolo nel carrello, il cliente può eseguire due operazioni:
 - ▶ richiedere una nuova quantità
 - ▶ rimuovere l'articolo dal carrello
- Inoltre il cliente può abbandonare la pagina del carrello in qualunque momento

- **Nome:** AggiornaCarrello
- **ID:** UC2
- **Attori:** Cliente
- **Precondizioni:** Il contenuto del carrello è visibile
- **Sequenza principale:**
 - ▶ 1. Il caso d'uso inizia quando il cliente seleziona un articolo nel carrello
- **Sequenze alternative:**

come possiamo migliorarlo...

- Possibili ramificazioni, dopo aver selezionato un articolo nel carrello:
 - ▶ se il cliente richiede una nuova quantità il sistema aggiorna la quantità di quell'articolo
 - ▶ se il cliente seleziona 'rimuovi articolo' il sistema elimina quell'articolo dal carrello
- Sequenza alternativa:
 - ▶ In qualunque momento il cliente abbandona la pagina del carrello

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

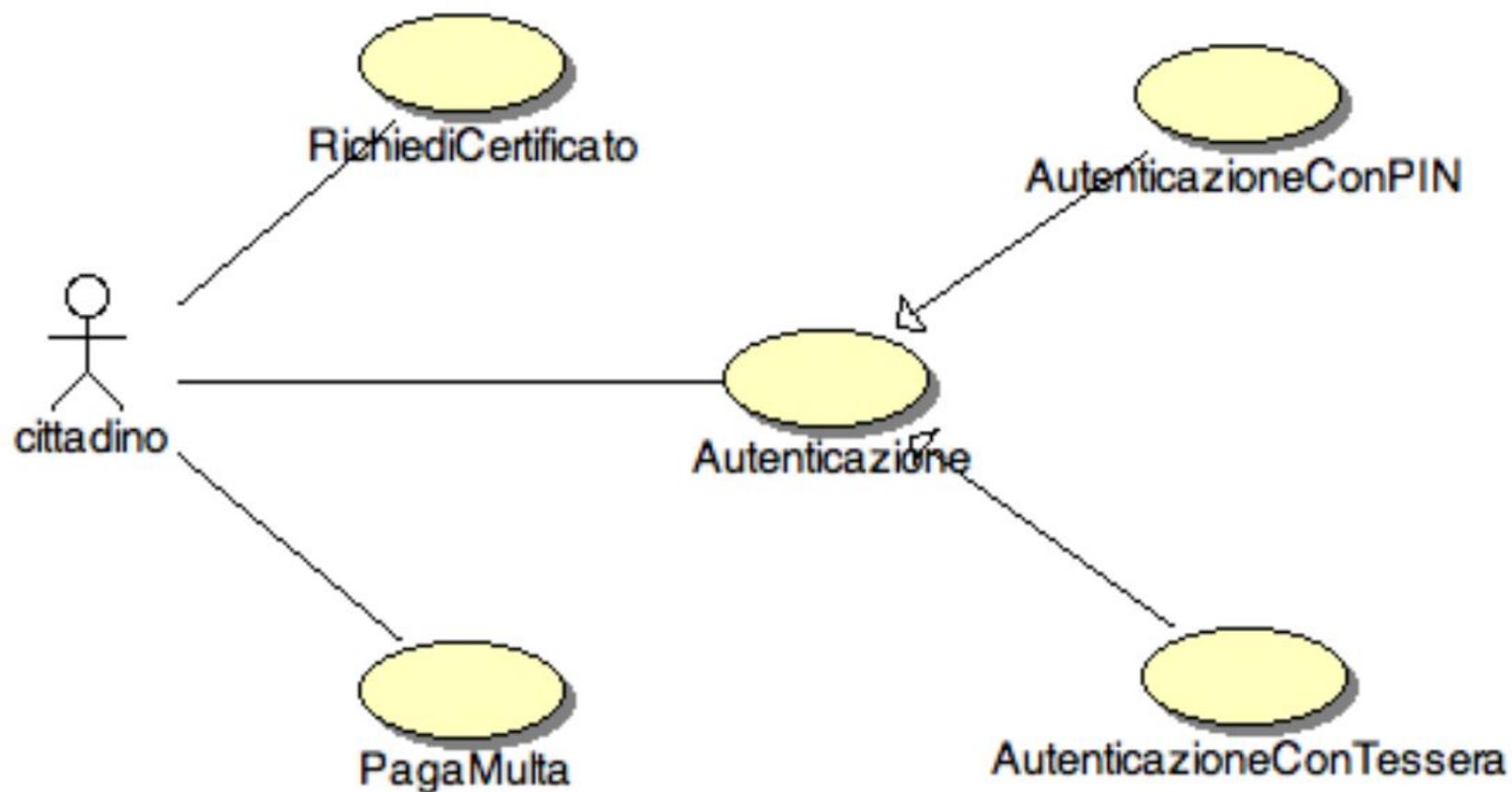
Sequenza alternativa 1:

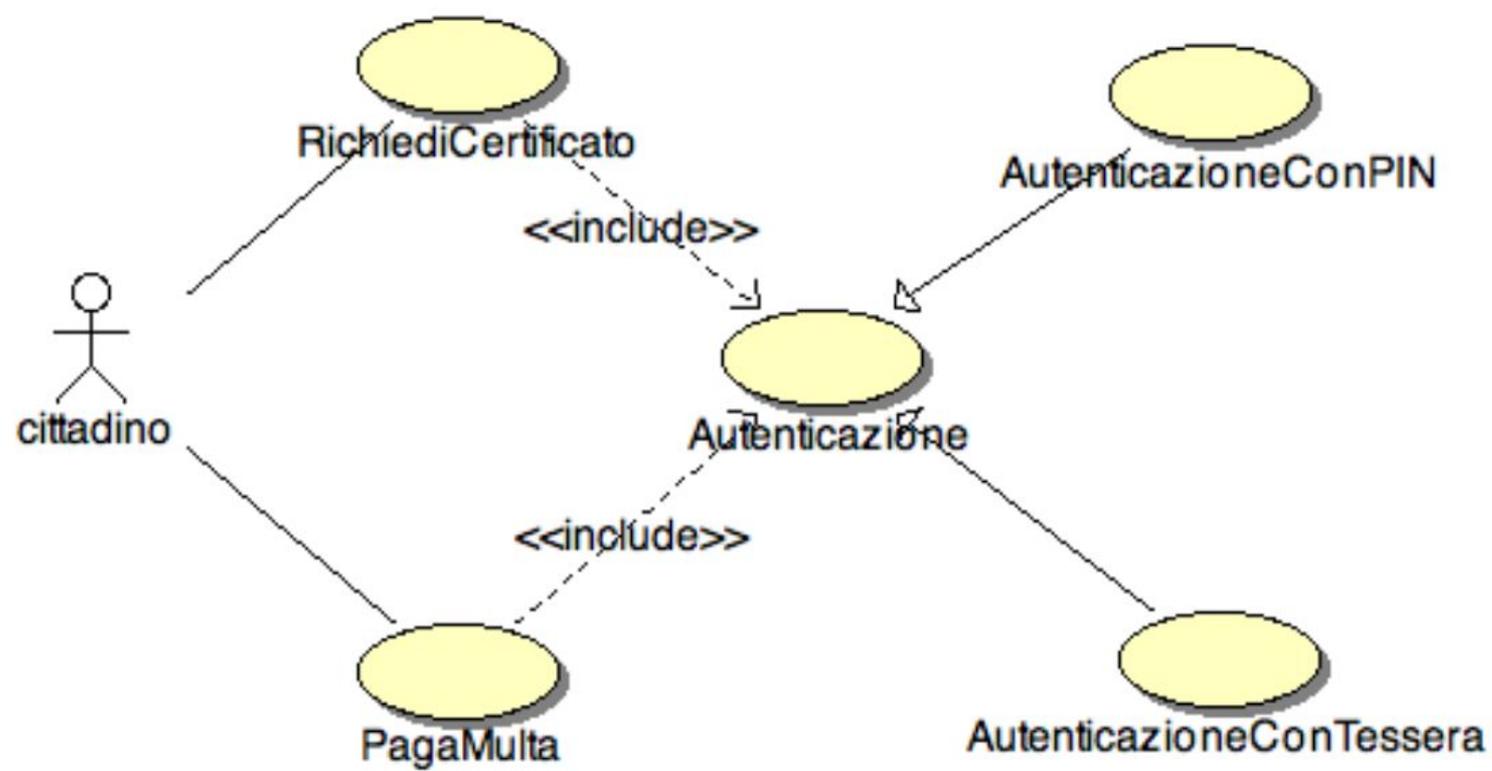
1. In qualunque momento il Cliente può abbandonare la pagina del carrello

Postcondizioni:

Esercizio 5: 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 6: Museo → Ritira bagaglio

- Si consideri l'esercizio relativo al museo. Scrivere la specifica del caso d'uso 'Ritira biglietto'.

- **Nome:** RitiraBiglietto
- **ID:** CU5
- **Precondizioni:** l'utente ha acquistato il biglietto
- **Sequenza principale:**
 - ▶ 1. Il caso d'uso inizia quando il cliente seleziona 'ritira' alla biglietteria automatica
 - ▶ 2. L'utente specifica gli estremi del biglietto
 - ▶ 3. Il sistema registra la consegna
 - ▶ 4. Il sistema eroga il biglietto
- **Sequenze alternative:**
 - ▶ Al punto 1, se gli estremi non sono validi: re-immettere i dati.
 - ▶ Al punto 1, se il biglietto è stato già erogato: esci.

Acquista Bevanda

Nome: AcquistaBevandaConContanti

Breve descrizione: L'utente prende una bevanda calda pagando in contanti

Attore Principale: Utente

Attore Secondario: nessuno

Precondizione: nessuna

Postcondizione: L'utente ha raccolto la bevanda calda ed è stato reso eventuale resto oppure è segnalato che mancano soldi

S.P.E.:

1. L'utente inserisce tante monete quanto il prezzo indicato sul pannello del distributore per la bevanda che ha intenzione di prendere
2. **Include** SelezioneErogazione
3. Restituisce eventuale resto

Nome: SelezioneErogazione

Breve descrizione: scelta bevanda ed erogazione

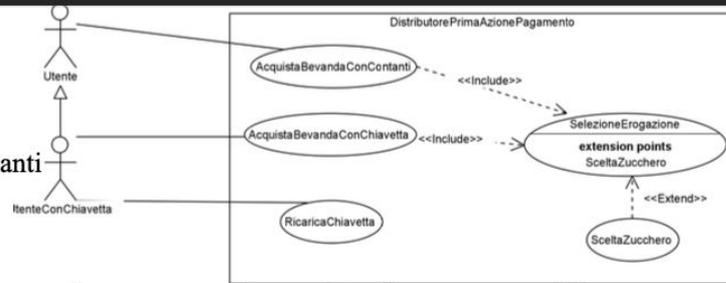
Attore Principale: nessuno, Attore Secondario: nessuno

Precondizione: pagamento effettuato oppure chiavetta inserita

Postcondizione: L'utente ha raccolto la bevanda calda oppure è segnalato che mancano soldi

S.P.E.:

1. L'utente sceglie la bevanda tra quelle disponibili
2. Il sistema controlla che il credito sia sufficiente
3. Se (credito insufficiente)
 - 3.1 il sistema lo segnala
4. Altrimenti
 - 4.1 Il sistema fa scendere il bicchiere
 - 4.2 Se (scelta Zucchero)
 - L'utente indica la quantità di zucchero
 - 4.3 Il sistema eroga la bevanda
 - 4.4 L'utente raccoglie la bevanda



Consigli nell'individuazione dei casi d'uso

- **Mantenere i casi d'uso brevi e semplici**
 - ▶ la descrizione non dovrebbe superare una pagina
 - ▶ evitare dettagli di progettazione
 - ▶ non appesantirli con informazioni non essenziali
- **Evitare la scomposizione funzionale**
 - ▶ non scomporre i casi d'uso con il metodo top-down (es. caso d'uso GestisciBiblioteca scomposto in GestioneLibri e GestionePrestiti e via via nei dettagli)
 - ▶ i casi d'uso emergono dai requisiti, non bisogna cercare di organizzarli in maniera artificiosa

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

**Esercitazione su
analisi dei requisiti
“House of cars”**