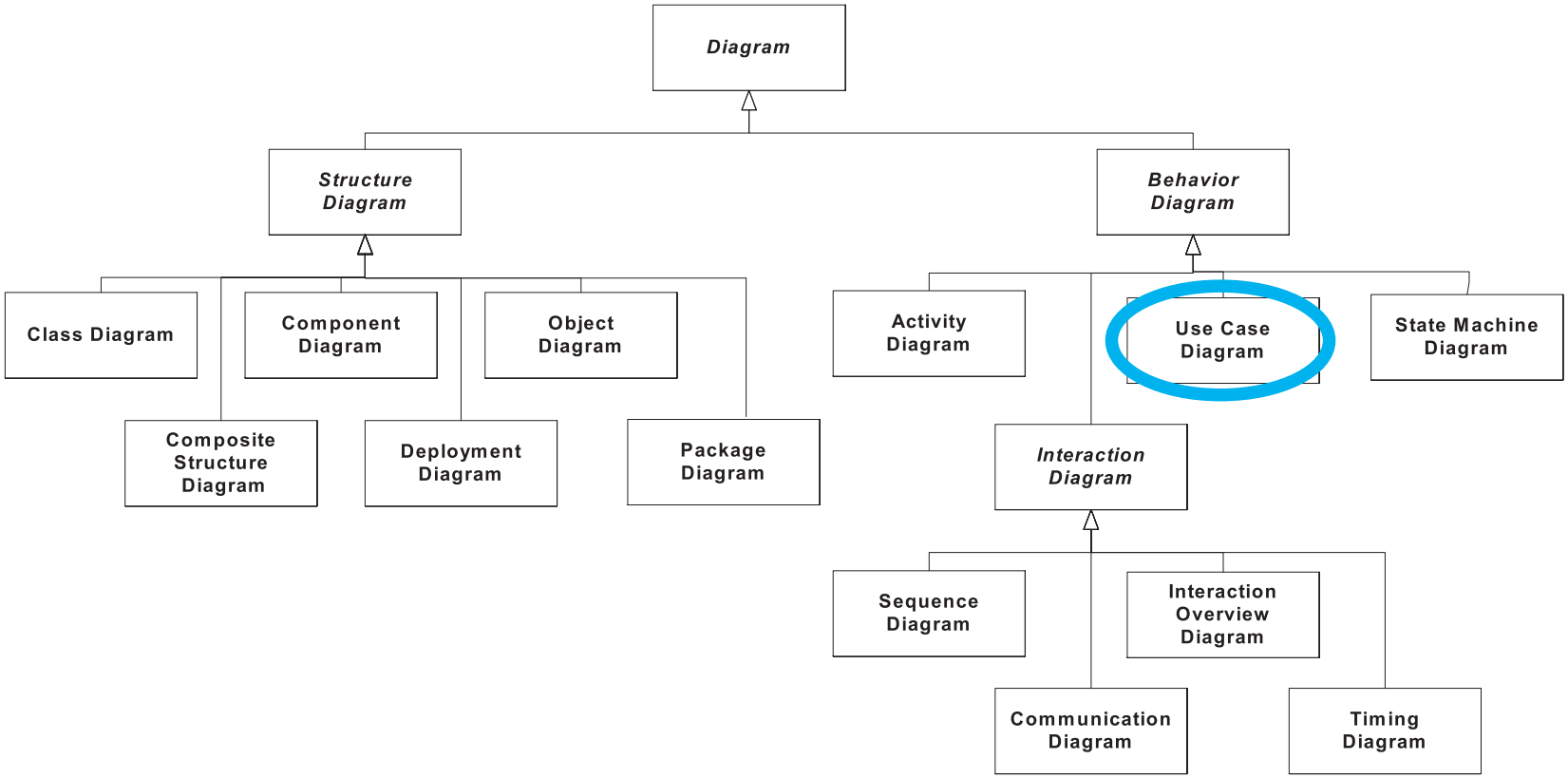
Il diagramma dei casi d'uso

Laboratorio di Sistemi e Processi Organizzativi
Gian Piero Favini

A.A. 2006-2007

Tassonomia dei diagrammi UML 2



Che cos'è e a cosa serve

- Si tratta di un diagramma che esprime un comportamento, offerto o desiderato, sulla base dei suoi risultati osservabili.
- L'oggetto esaminato è solitamente un sistema o una sua parte.
- Individua chi o che cosa ha a che fare con il sistema (*attore*) e che cosa viene fatto (*caso d'uso*).
- Si tratta tipicamente del primo tipo di diagramma ad essere creato in un processo o ciclo di sviluppo, nell'ambito dell'analisi dei requisiti.
- Modella i *requisiti funzionali* di un sistema.

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 sono particolarmente utili se la maggior parte dei requisiti sono funzionali.

Pitfalls

- Questo diagramma specifica *cosa* va fatto, non *come* va fatto. Spetta alle fasi successive capire come realizzare i casi d'uso, la fase dei requisiti deve solo specificarli.
- Evitare riferimenti a tecnologie specifiche nei casi d'uso, per quanto possibile.
- Scegliere il giusto livello di dettaglio:
  - una granularità troppo fine rischia di sconfinare nella progettazione
  - una troppo grossa rischia di generare ambiguità nelle fasi successive
- Tenere in mente la leggibilità, questo diagramma dovrebbe risultare comprensibile anche a un non-esperto:
  - in particolare il cliente (è uno strumento per chiarirsi)

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.

Desiderata: esempio (1)

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?

Desiderata: esempio (2)

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...

Desiderata riformulati in maniera migliore.

Estrarre i requisiti

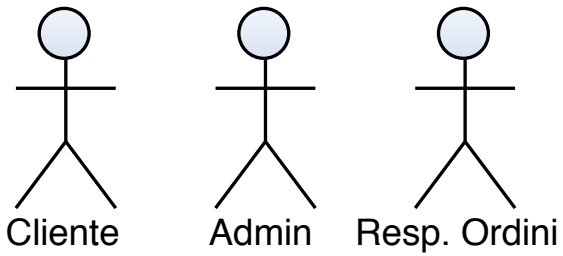
Chi interagisce con il sistema (attori)?

- Clienti
- Amministratori del negozio online
- Reparto ordini

Cosa fanno (casi d'uso)?

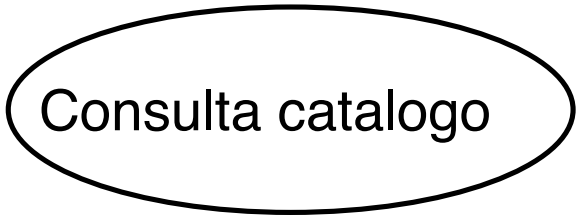
- Il cliente si registra, consulta il catalogo ed effettua acquisti
- L'amministratore organizza il catalogo, che è diviso in categorie
- Il reparto ordini riceve ordini da evadere

Elementi del diagramma: attore



- Un attore specifica un ruolo assunto da un utente o altra entità che interagisce con l'argomento del diagramma nell'ambito di un'unità di funzionamento (caso d'uso).
- Un attore è esterno all'argomento (sistema) oggetto del diagramma.
- Non è necessariamente umano: oggetti fisici, agenti e componenti software, condizioni ambientali, ...
- Una singola entità o parte di entità può assumere molti ruoli, e una collezione di entità può essere rappresentata da un unico attore.

Elementi del diagramma: caso d'uso (1)

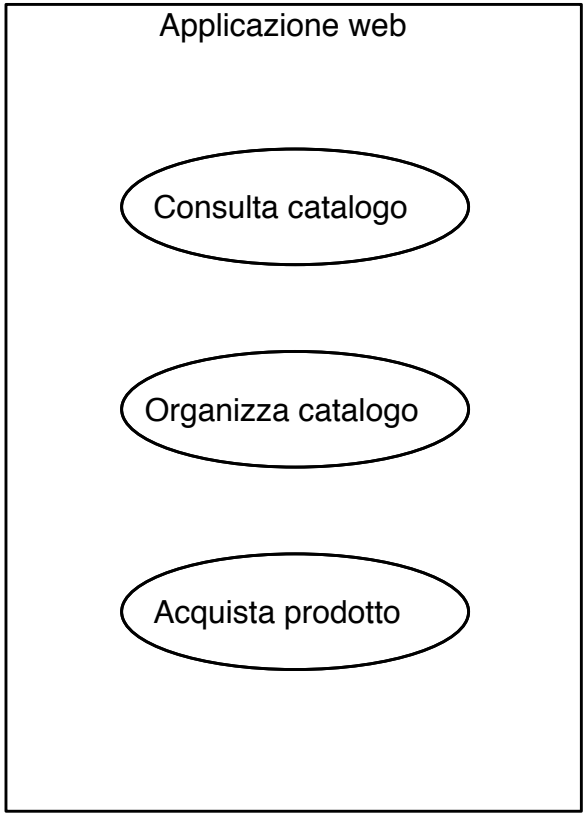


- Un caso d'uso specifica un insieme di azioni che producono un risultato osservabile per uno o più attori.
- Si tratta di un'unità coerente di funzionamento che il sistema fornisce ad uno o più utenti.
- La descrizione all'interno dovrebbe essere basata su un verbo o su un sostantivo esprimente un avvenimento.

Elementi del diagramma: caso d'uso (2)

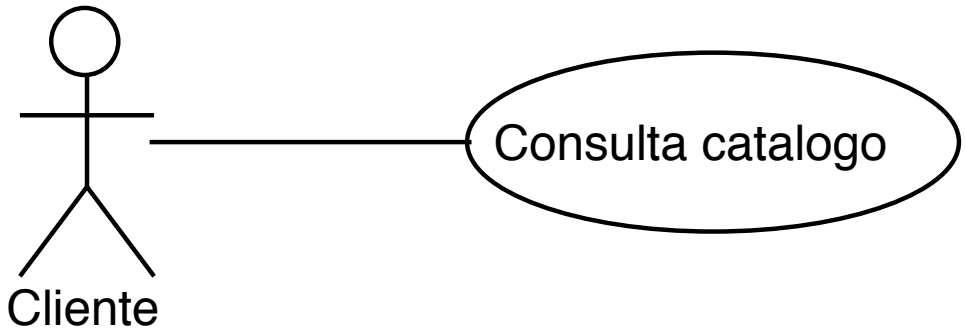
- Un caso d'uso è sempre iniziato da un attore: in UML, un evento è sempre legato all'entità che lo ha generato.
- L'attore che inizia un caso d'uso è detto *primario*, gli altri attori che interagiscono nell'ambito di quel caso d'uso sono *secondari*.
- Un caso d'uso è un *classificatore dotato di comportamento*: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

Elementi del diagramma: sistema



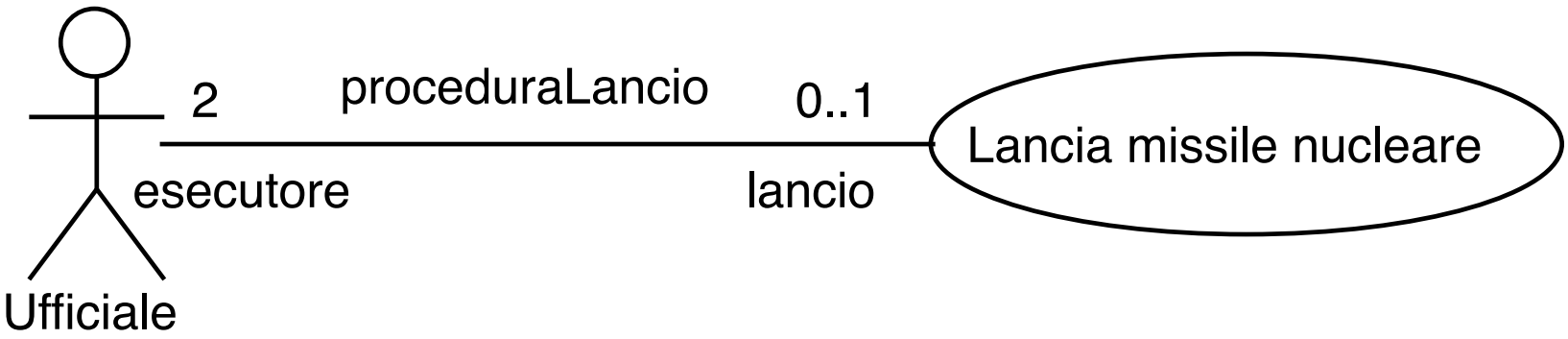
- Delimita l'argomento del diagramma, specificando i confini del sistema.

Elementi del diagramma: associazione (1)



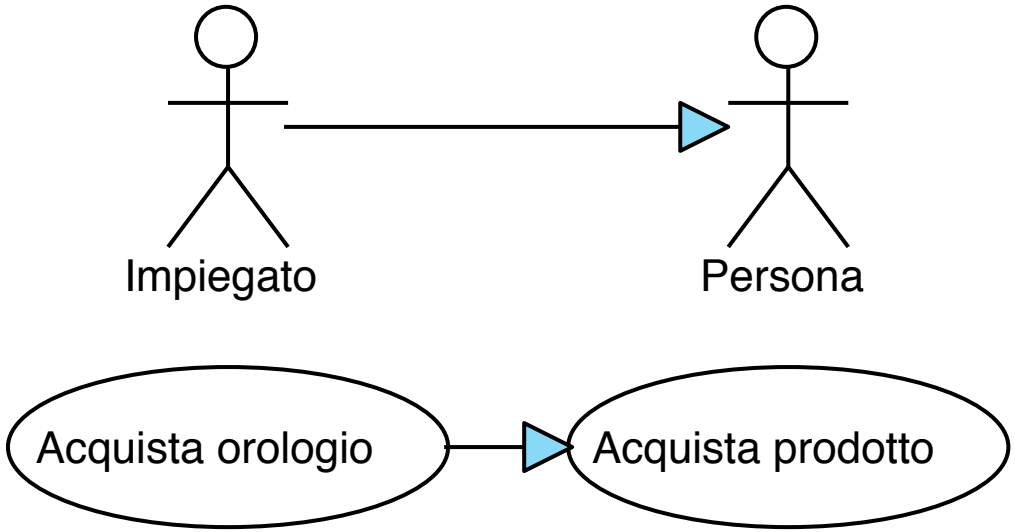
- Collega gli attori ai casi d'uso.
- Un attore si può associare solo a casi d'uso, classi e componenti.
- Un caso d'uso non si può associare ad altri casi d'uso riguardanti lo stesso argomento.
- Entrambi supportano solo associazioni binarie.

Elementi del diagramma: associazione (2)



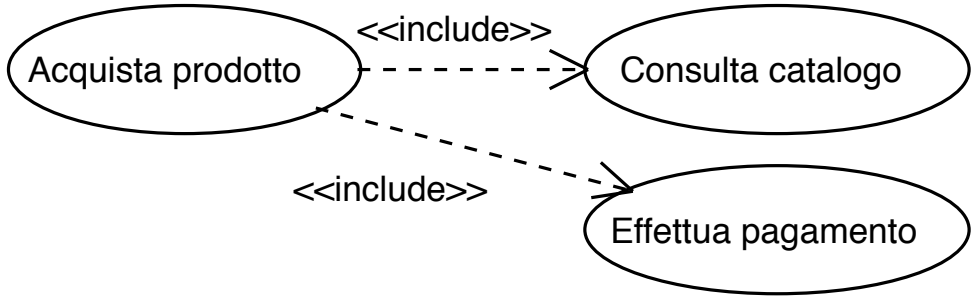
- Alcune caratteristiche opzionali comuni a tutte le associazioni in UML:
  - nome
  - molteplicità
  - ruoli

Elementi del diagramma: generalizzazione



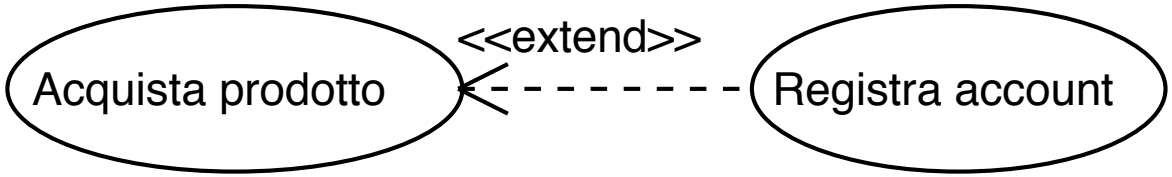
- Collega un attore o caso d'uso ad un altro più generale.
- Il figlio può sostituire il genitore dovunque questi appaia.
- In UML, elementi astratti (che non possono essere istanziati) hanno il nome in corsivo.

Elementi del diagramma: include



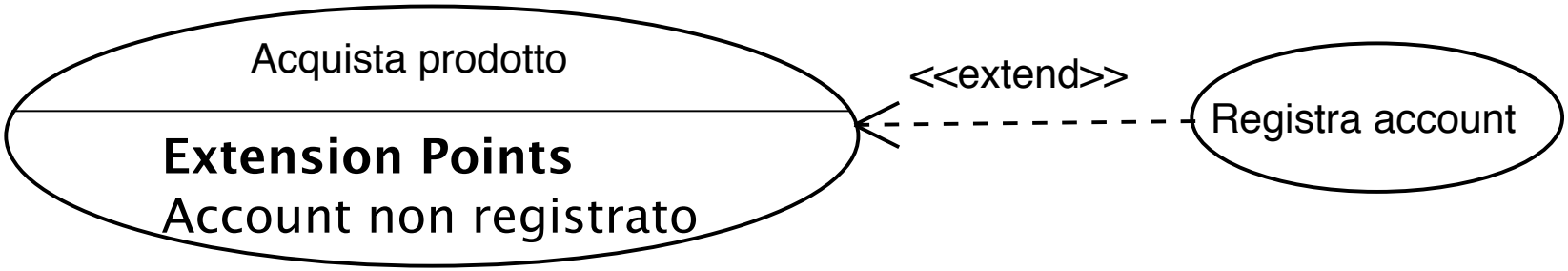
- Una dipendenza tra casi d'uso; il caso incluso fa parte del comportamento di quello che lo include.
- L'inclusione *non* è opzionale ed avviene in ogni istanza del caso d'uso.
- La corretta esecuzione del caso d'uso che include dipende da quella del caso d'uso incluso.
- Non si possono formare cicli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

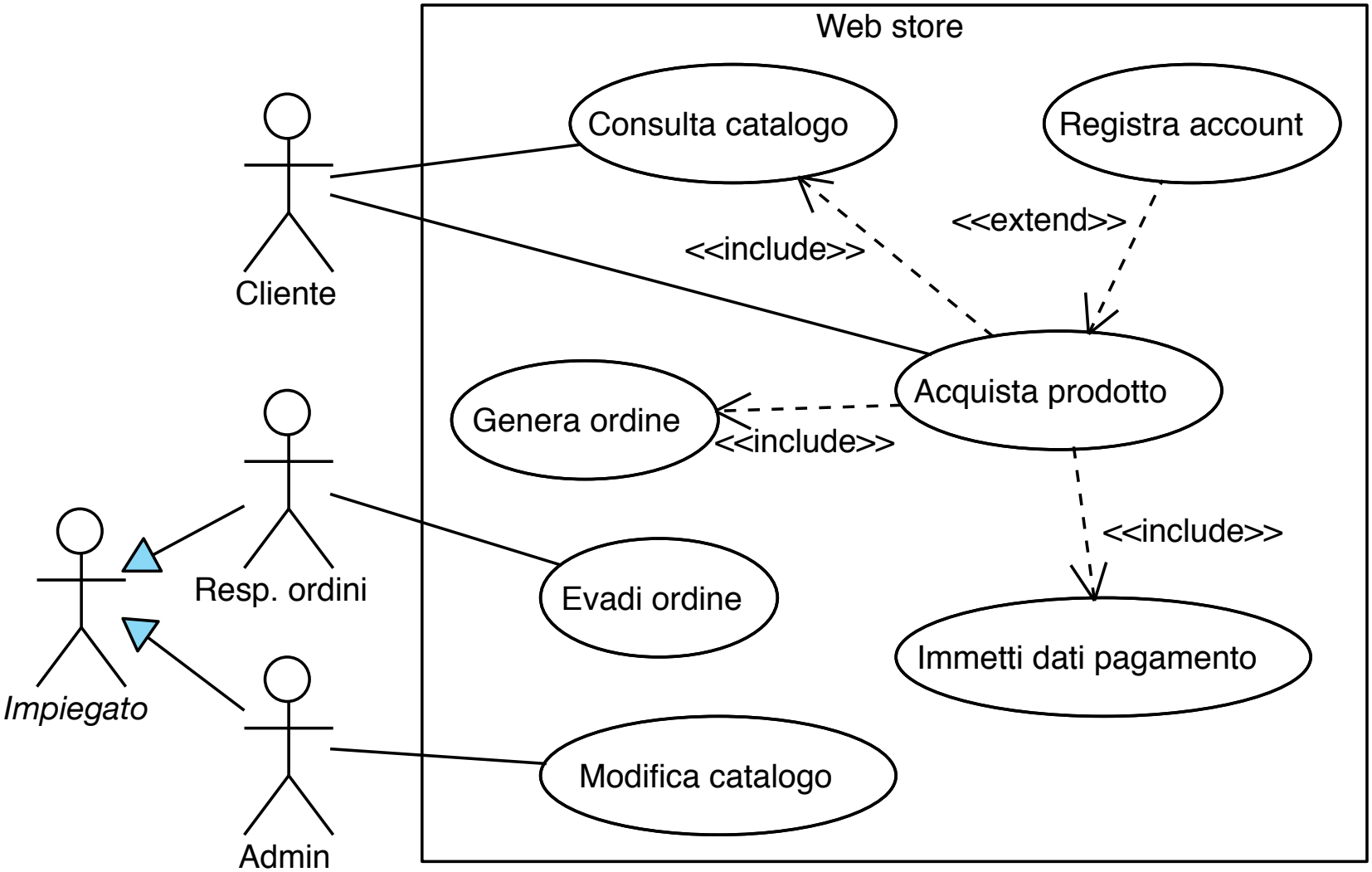


- Un caso d'uso raggiunto da almeno un extend può opzionalmente visualizzare i propri extension points.
- Specifica i punti e/o condizioni dell'esecuzione in cui il comportamento viene esteso.
- Se gli extension points sono molti, alcuni tool possono supportare la rappresentazione a rettangolo (i casi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

include vs. extend

- Include specifica comportamento *obbligatorio*.
- Extend specifica comportamento *supplementare*.
- 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



Modellare i casi d'uso

- Se i requisiti del sistema non sono banali, 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 dei comportamenti
  - lo stato del sistema e degli attori prima e dopo l'esecuzione del caso d'uso
- Altri diagrammi (attività, stato, interazione) si occupano di queste viste, ma devono partire da una specifica.

Specifiche del caso d'uso

- Ogni caso d'uso ha un nome e una specifica.
- La specifica è composta da:
  - *precondizioni (entry conditions):* condizioni che devono essere vere prima che il caso d'uso possa essere eseguito (sono vincoli sullo stato iniziale del sistema)
  - *sequenza degli eventi (flow of events):* i passi che compongono il caso d'uso
  - *postcondizioni (exit conditions):* condizioni che devono essere vere quando il caso d'uso termina l'esecuzione

Esempio specifica caso d'uso

Nome del caso d'uso	**Caso d'uso: PagamentoIVA**
Identificatore univoco	**ID:** UC1
Gli attori interessati dal caso d'uso	**Attori:** Tempo, Fisco
Lo stato del sistema prima che il caso d'uso possa iniziare	**Precondizioni:** 1. Si è concluso un trimestre fiscale
I passi del caso d'uso	**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 sitema trasmette un pagamento elettronico al Fisco.
Lo stato del sistema quando l'esecuzione del caso d'uso è terminata	**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>`

Esempi

- ~~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

Ramificazione di una sequenza

- UML usa parole chiave per esprimere ramificazione, ripetizione o sequenze alternative.
- È bene non eccedere con le ramificazioni.
- Parola chiave **Se**: indica una ramificazione della sequenza degli eventi.
- Sequenze alternative: ramificazioni che non possono essere espresse utilizzando il Se. Ad esempio ramificazioni dovute a condizioni che si possono verificare in un qualunque momento.
- Ripetizioni all'interno di una sequenza:
  - parola chiave **Per** (For)
  - parola chiave **Fintantoché** (While)

Esempio ramificazione

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:

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.
- Uno scenario è un'istanza di un caso d'uso.
- 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
- Utili per immaginare il sistema all'opera e modellare i casi di test.

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.

Consigli per l'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

Requisiti: in pratica

- 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.