

Ingegneria del Software I

Unified Modeling Language

Use Case Diagram

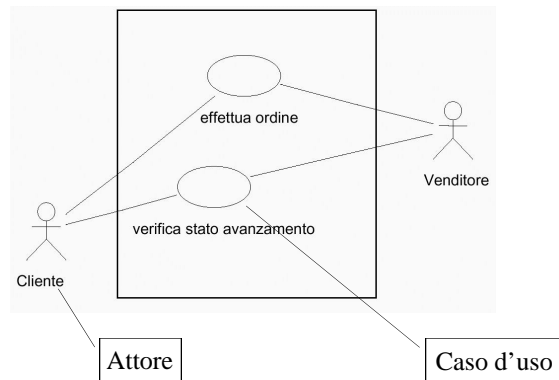
1

Requisiti e casi d'uso

- ◆ Il primo passo di “qualsiasi” processo di sviluppo è la definizione dei requisiti
 - Definizione del Business Model
 - Solitamente informale e in linguaggio naturale
- ◆ Jacobson (OOSE) propone una notazione particolare che è confluita in UML: i casi d'uso
 - Esprime una interazione tipica tra un utente ed il sistema software. Il punto di vista da adottare nella definizione dei casi d'uso è quello dell'attore, non quello delle funzionalità interne al sistema.
 - Cattura il comportamento esterno del sistema da sviluppare, senza dover specificare come tale comportamento viene realizzato (sistema = black-box)
 - Forniscono una descrizione delle “modalità” di utilizzo del sistema da parte di un utilizzatore (attore)

2

Esempio: gestione ordini



3

Casi d'uso per descrivere le interazioni utente-sistema

- ◆ I casi d'uso possono essere descritti sotto forma di scenario di interazione (dialogo) tra gli utilizzatori e il sistema:
 - Caso d'uso: effettuare ordine
 - » il cliente richiede l'elenco dei prodotti
 - » il sistema propone i prodotti disponibili
 - » il cliente sceglie i prodotti che desidera
 - » il sistema fornisce il costo totale dei prodotti selezionati
 - » il cliente conferma l'ordine
 - » il sistema comunica l'accettazione dell'ordine
- ◆ Un caso d'uso rappresenta una funzionalità dal punto di vista di chi la utilizza
 - ogni caso d'uso può soddisfare più requisiti funzionali
 - un requisito funzionale può dare origine a più casi d'uso
 - a ogni caso d'uso possono venire associati più requisiti non funzionali

4

UML - Use Case Diagram

Ingegneria del Software I

Elementi grafici del modello dei casi d'uso

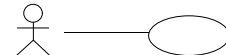
Il modello dei casi d'uso presenta una vista del sistema con gli *attori*, i *casi d'uso* e le loro *associazioni*



Attore



Caso d'uso



Relazione di Associazione

5

Attore

- ◆ Un ruolo (o un insieme di ruoli) che l'utente del caso d'uso svolge nell'interagire col sistema
 - E' esterno al sistema
 - Può essere:
 - » Una classe di persone fisiche (es. Fornitore)
 - » Un altro sistema software (es. Sistema di contabilità)
 - » Un dispositivo hardware esterno (es. Sensore)
 - Controlla le funzionalità
 - Fornisce input o riceve output dal sistema
 - Può partecipare a più use case
 - Uno stesso utente può rivestire più ruoli nel medesimo use case
 - Uno stesso use case può coinvolgere più actor
 - Un actor non necessariamente è un essere umano
- ◆ Un attore sollecita il sistema con un qualche evento e ne riceve un output.

6

Caso d'Uso

- ◆ Descrive il comportamento del sistema quando sollecitato da un attore
- ◆ Il comportamento è descritto in maniera testuale, come **sequenza di transazioni** del sistema, il cui compito è produrre un risultato di valore misurabile per un attore del sistema.
 - Una transazione è un insieme atomico di attività (sono completate o non sono eseguite affatto)
 - La descrizione di un caso d'uso definisce cosa accade nel sistema in seguito all'evento di innesco
 - Generalmente lo stimolo parte dall'attore, ma può anche essere il sistema stesso ad iniziare il caso d'uso (es. Produzione cedolini a fine mese, ricarica automatico di un magazzino)
 - Un caso d'uso corrisponde ad un compito che l'attore vuole eseguire (l'attore inizia il caso d'uso) o il sistema deve eseguire (il sistema inizia il caso d'uso)

7

Casi d'uso e transazioni

Esempio:



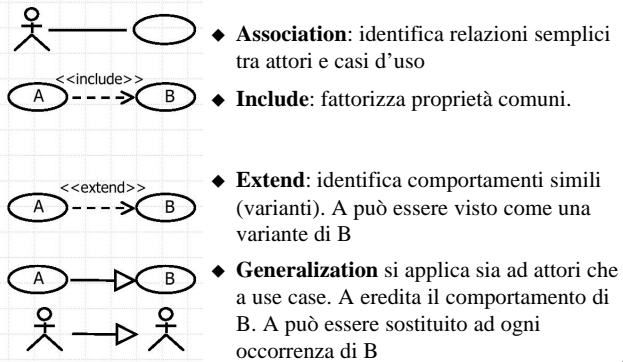
Transazioni:

- verifica esistenza cliente in anagrafica
- acquisizione nuova anagrafica
- acquisizione firma digitalizzata
- inserimento nuovo conto corrente

8

Ingegneria del Software I

Relazioni principali



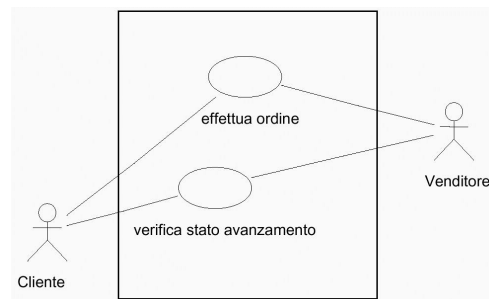
9

Diagramma dei Casi d'Uso

- ◆ Contiene
 - Attori
 - Casi d'uso
 - Relazioni di associazione, dipendenza e generalizzazione
- ◆ Usato per modellare il contesto di un sistema
 - Gli attori sono esterni al sistema
 - I casi d'uso sono all'interno del sistema
- ◆ Usato per modellare (visualmente) i requisiti funzionali
 - Ogni caso d'uso corrisponde a uno o più requisiti funzionali
 - Ogni caso d'uso è coinvolto in una qualche relazione
- ◆ Diversi livelli di dettaglio
 - Decomposizione gerarchica

10

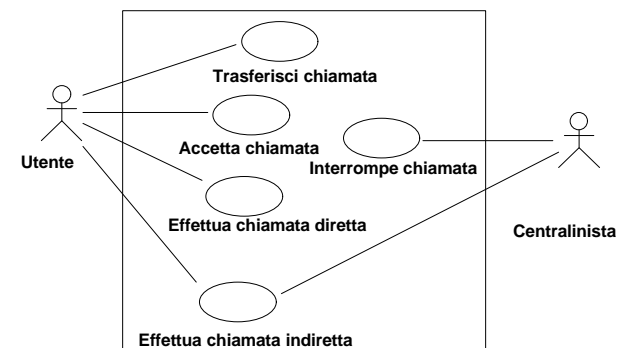
Esempio: gestione ordini



Lo Use Case diagram di livello più alto definisce il contesto del modello

11

Esempio: centralina telefonica per ufficio

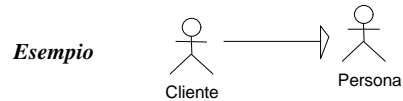


12

Ingegneria del Software I

Attori e generalizzazione

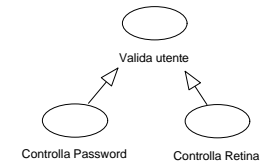
Tra gli actor possono esistere delle relazioni di generalizzazione



13

Generalizzazione fra casi d'uso

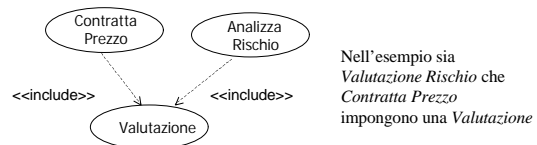
- ◆ Un caso d'uso figlio eredita il comportamento ed il significato del caso d'uso padre
- ◆ Il figlio può aggiungere o modificare il comportamento del padre (es. Caso d'uso Valida Utente)



14

Relazione di Inclusione

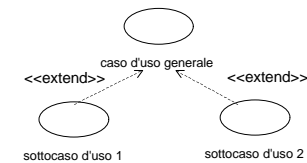
- ◆ Un comportamento comune a più casi d'uso diventa un caso d'uso che è incluso nei casi d'uso di partenza.
 - L'inclusione evita di copiare la descrizione del comportamento comune nei differenti use case che lo coinvolgono.
 - L'inclusione avviene in un punto preciso della descrizione del caso d'uso includente. Il caso incluso non ha senso da solo.
 - Rappresentato graficamente come una dipendenza come `<<include>>`



15

Relazione di Estensione

- ◆ Comportamenti alternativi o eccezionali (opzionali) di un caso d'uso formano dei casi d'uso che estendono il caso d'uso generale
 - nel caso d'uso esteso viene inserito un punto d'estensione
 - nei sottocasi d'uso si fa riferimento a questi punti
- ◆ La relazione `extends` va utilizzata quando abbiamo uno use case simile ad un altro che però fa qualcosa in più.



16

Ingegneria del Software I

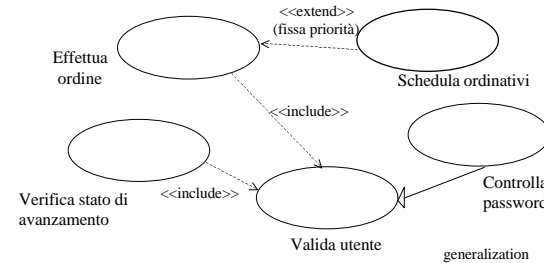
Extends versus Includes

Una semplice regola:

- ◆ utilizzare extends quando si descrive una variazione al comportamento normale;
- ◆ utilizzare includes quando si sta ripetendo la descrizione di un comportamento presente già in uno o più differenti use cases.

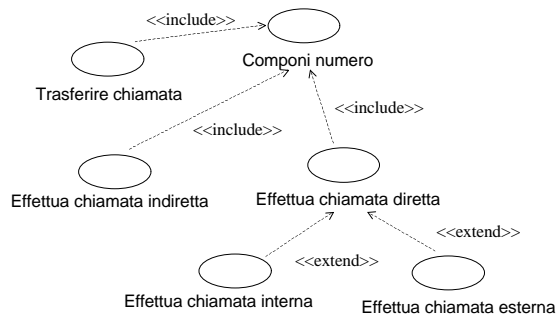
17

Esempio: gestione ordini



18

Esempio: centralina telefonica



19

Documentazione: descrizione dei casi d'uso

◆ Per ogni use case:

- **Titolo:** Effettua chiamata diretta
- **Autori:** Pluto e Topolino
- **Descrizione:** L'utente compone il numero per effettuare una chiamata diretta
- **Attori:** Utente
- **Pre-condizioni:** Non identificate
- **Post-condizioni:** Non identificate
- **Scenari**

20

Ingegneria del Software I

Scenari ...

- ◆ Uno scenario è un'istanza di uno use case
 - È una "esecuzione" particolare dello use case
 - Rappresenta il comportamento (le azioni e gli eventi) del sistema nel caso particolare considerato
 - Definisce cosa accade nel sistema in seguito all'evento di innesco
 - » come e quando il caso d'uso inizia
 - » chi inizia il caso d'uso
 - » interazione tra attore/i e caso d'uso e cosa viene scambiato
 - » come e quando c'è bisogno di dati memorizzati o di memorizzare dati
 - » come e quando il caso d'uso termina
- ◆ Stili di descrizione
 - In linguaggio naturale con chiaro flusso da eseguire
 - UML propone una notazione diagrammatica particolare: gli Interaction Diagrams

... scenari

- ◆ Ogni use case dovrebbe essere corredato da un insieme di scenari
 - Scenari principali (più possibile)
 - » Tutto funziona correttamente
 - Scenari secondari (pochi e significativi)
 - » Eccezioni (eventuali problemi o malfunzionamenti)
- ◆ Si devono definire tanti scenari quanti sono quelli necessari per capire il corretto funzionamento del sistema e le eccezioni che si ritengono significative durante l'analisi

22

Caso d'uso: Effettua chiamata interna

Scenario 1: chiamata effettuata con successo

- ◆ L'utente solleva la cornetta
- ◆ Il tono di libero interno suona
- ◆ L'utente compone il numero
- ◆ La chiamata è inoltrata sulla rete telefonica
- ◆ Il telefono chiamato squilla
- ◆ Il tono di attesa suona
- ◆ L'utente chiamato risponde
- ◆ I telefoni sono connessi
- ◆ Il telefono chiamato cessa di squillare
- ◆ Il tono di attesa finisce
- ◆ L'utente chiamato ripone la cornetta sul telefono
- ◆ I telefoni sono disconnessi
- ◆ L'utente ripone la cornetta sul telefono

23

Caso d'uso: Effettua chiamata interna

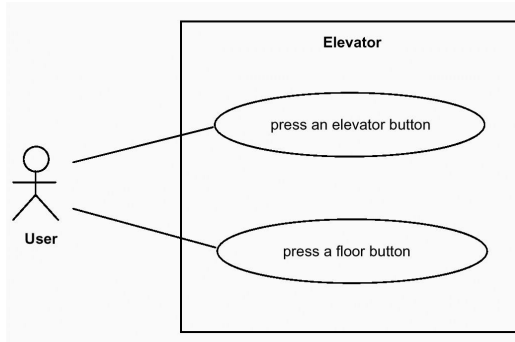
Scenario 2: il telefono chiamato è occupato

- ◆ L'utente solleva la cornetta
- ◆ Il tono di libero interno suona
- ◆ L'utente compone il numero
- ◆ La chiamata è inoltrata sulla rete telefonica
- ◆ Il telefono chiamato è occupato
- ◆ Il tono di occupato suona
- ◆ L'utente ripone la cornetta sul telefono

24

Ingegneria del Software I

Elevator problem: use case diagram



25

Elevator problem: normal scenario

1. User A presses Up floor button at floor 3 to request elevator. User A wishes to go to floor 7.
2. Up floor button is turned on.
3. An elevator arrives at floor 3. It contains User B who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4. Up floor button is turned off.
5. Elevator doors open. User A enters elevator.
6. User A presses elevator button for floor 7.
7. Floor 7 elevator button is turned on.
8. Elevator doors close.
9. Elevator travels to floor 7.
10. Floor 7 elevator button is turned off.
11. Elevator doors open to allow User A to exit elevator.
12. Timer starts. User A exits.
13. Elevator doors close after timeout.
14. Elevator proceeds to floor 9 with User B.

26

Elevator problem: abnormal scenario

1. User A presses Up floor button at floor 3 to request elevator. User A wishes to go to floor 1.
2. Up floor button is turned on.
3. An elevator arrives at floor 3. It contains User B who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4. Up floor button is turned off.
5. Elevator doors open. User A enters elevator.
6. User A presses elevator button for floor 1.
7. Floor 1 elevator button is turned on.
8. Elevator doors close after timeout.
9. Elevator travels to floor 9.
10. Floor 9 elevator button is turned off.
11. Elevator doors open to allow User B to exit elevator.
12. Timer starts. User B exits.
13. Elevator doors close after timeout.
14. Elevator proceeds to floor 1 with User A.

27

Costruzione del modello dei casi d'uso (1)

- ◆ Gli use case possono essere ricavati dalle interviste con gli utenti. Si identificano
 - Gli obiettivi: ciò che il sistema dovrebbe fare secondo gli utenti
 - Le interazioni: cosa vorrebbero (potrebbero) fare i diversi utenti
- ◆ Gli use case di alto livello sono volutamente generici
 - I dettagli vanno aggiunti raffinando le funzionalità del sistema
 - Bisogna esplorare le diverse possibilità per non introdurre prematuramente scelte progettuali

28

Ingegneria del Software I

Costruzione del modello dei casi d'uso (2)

- ◆ Il processo di definizione degli use case è iterativo
 - Si inizia identificando il comportamento più semplici
 - Si descrivono i comportamenti alternativi e più complessi
- ◆ Quando smettere?
 - Un buon livello di dettaglio facilita tutte le attività successive
 - Troppi dettagli
 - » Complicherebbero inutilmente la descrizione
 - » Introdurrebbero prematuramente scelte progettuali
 - » Precluderebbero la visione d'insieme

29

Passi per la costruzione di un modello dei casi d'uso

1. Identificazione degli attori
2. Identificazione dei casi d'uso
3. Definizione delle associazioni fra attori e casi d'uso
4. Descrizione dei casi d'uso
5. Strutturazione dei casi d'uso

30

1. Identificazione degli attori

- ◆ Identificare le persone che interagiscono con il sistema per eseguire qualche compito
 - persone che necessitano del sistema per svolgere qualche compito
 - persone che il sistema richiede per svolgere qualche compito
 - considerare sia i compiti principali che quelli di supporto al sistema, quali manutenzione ed amministrazione
- ◆ Raggruppare le persone identificate secondo i loro ruoli (responsabilità) rispetto al sistema
- ◆ Identificare altri sistemi software e dispositivi esterni che interagiscono con il sistema per svolgere qualche compito

31

2. Identificazione dei casi d'uso

- ◆ Per ogni attore:
 - identificare i compiti o funzioni che l'attore deve essere in grado di eseguire
 - identificare i compiti che il sistema richiede che l'attore esegua
- ◆ Raggruppare compiti e funzioni in casi d'uso
 - i casi d'uso devono corrispondere ad un obiettivo specifico per l'attore o per il sistema
 - raggruppare funzioni che sono eseguite in sequenza o che sono innescate dallo stesso evento
 - il caso d'uso non deve essere né troppo grande né troppo piccolo (non decomporre funzioni complesse in casi d'uso per ogni sottofunzione)
- ◆ Assegnare al caso d'uso un nome significativo che sintetizzi la/le funzionalità svolta/e

32

Ingegneria del Software I

3. Definizione delle associazioni fra attori e casi d'uso

- ◆ Ogni attore deve partecipare in almeno un caso d'uso
- ◆ Ogni caso d'uso deve avere almeno un attore con cui comunica
- ◆ Se due attori partecipano agli stessi casi d'uso considerare la possibilità di combinarli in un unico attore

33

4. Descrizione dei casi d'uso

- ◆ Considerare sia lo scenario principale che scenari alternativi ed eccezionali
- ◆ Per ogni scenario specificare:
 - come e quando il caso d'uso inizia
 - chi avvia il caso d'uso
 - interazione tra attore/i e caso d'uso e cosa viene scambiato
 - come e quando c'è bisogno di dati memorizzati o di memorizzare dati
 - come e quando il caso d'uso termina
- ◆ Se due casi d'uso hanno comportamenti leggermente diversi e gli stessi attori, considerare la possibilità di avere un unico caso d'uso con scenari alternativi

34

5. Strutturazione dei casi d'uso

- ◆ Identificare le relazioni di generalizzazione ed estensione
 - specializzare i casi d'uso che hanno molti scenari alternativi
 - collegare i nuovi casi d'uso a quelli di partenza mediante relazioni di generalizzazione o di <<extend>>
- ◆ Identificare le relazioni di inclusione
 - individuare parti comuni in casi d'uso diversi
 - collegare i casi d'uso che condividono una parte comune al nuovo caso d'uso (rappresentante il comportamento condiviso) mediante l'associazione <<include>>

35

Oltre l'analisi dei requisiti

- ◆ Convalida del sistema
 - Gli use case possono essere utilizzati per ricavare i dati di test con cui convalidare il sistema
 - » Ogni use case rappresenta una funzionalità che andrebbe verificata
- ◆ Gestione del progetto
 - Gli use case propongono una “nuova” unità di misura
 - Gli use case potrebbero essere utili per
 - » Organizzare il progetto
 - » Stimare la complessità

36