Use case

Introduzione

Obiettivi:

- Identificare i *casi d'uso*
- Connettere i casi d'uso rispetto agli obiettivi dell'*attore principale*
- Mantenere uno stile *essenziale*, non troppo informativo, poichè potrebbe rendere maggiormente difficile l'individuazione del *goal*
- Adottare uno *sviluppo iterativo*

Use case spesso è indicato come il meccanismo per scoprire e registrare i requisiti, di carattere *funzionale*. La scrittura dei *casi d'uso* è un eccellente tecnica per capire e descrivere ciò che il *sistema* debba implementare, rispetto ad una totalità di influenze esterne che creino la situazione ideale per poter proseguire con il compimento dell'obiettivo principale da parte del *soggetto*. **UP**, ossia *Unified Process*, indica un certo modello dei casi d'uso, definito come **modello comportamentale**, per cui fortemente connesso rispetto ai *comportamenti* degli elementi trattati.

I *diagrammi* dei *casi d'uso* sono **diagrammi comportamentali** utilizzati per descrivere un insieme di azioni, o meglio *behavior*, che un sistema deve svolgere insieme ad uno o più utenti esterni al sistema. Inoltre, carattere imprescindibile di tale modellazione, prevede che ogni *use case* fornisca un **risultato di valore**, per gli utenti con cui interagiscono oppure per gli *stakeholders*. *Customers* e *final users* pongono certi obiettivi, detti *needs*, i quali richiedono che *sistemi* siano in grado di rendere chiaro il loro **connubio**.

Differenti sono gli approcci dettati, tuttavia le strategie migliori riguardano un uso semplice e famigliare degli strumenti messi a disposizione, poichè rendono più semplice la lettura del modello, contribuiscono alla ricerca dei requisiti necessari e abbassa il rischio di perdere l'obiettivo su cui fonda l'intera architettura, definito anche *missing the mark*.
Infatti il primo step ricade nella comprensione del problema assieme all'individuazione dei requisiti funzionali che comporranno i casi d'uso, affinchè siano comprensibili da parte dei portatori di interesse. Tuttavia, quest'ultimo passaggio rappresenta la maggiore difficoltà di questo modello, ossia la decisione e scoperta di cosa sia potenzialmente necessario; durante una prima fase di analisi potrebbero essere inseriti dei *layer sofistaticati-aggiuntivi*, i quali solitamente oscurano il reale proseguimento del *mark*.

Per cui, ricapitolando, il diagramma dei casi d'uso è utilizzato per specificare:

- I requisiti esterni degli stakeholders, funzionali rispetto a quanto il sistema debba implementare
- Le funzionalità offerte dal sistema per tutti i final user esterni
- Rispettare i limiti secondo cui il sistema debba sottostare, defininendo come tale ambiente debba interagire con il soggetto pur di proseguire verso il *path* voluto

Elementi del diagramma

In questa sezione si analizzano gli elementi comuni ad un qualsiasi diagramma dei casi d'uso, descrivendo successivamente caratteristiche uniche rispetto alle strategie elencate.

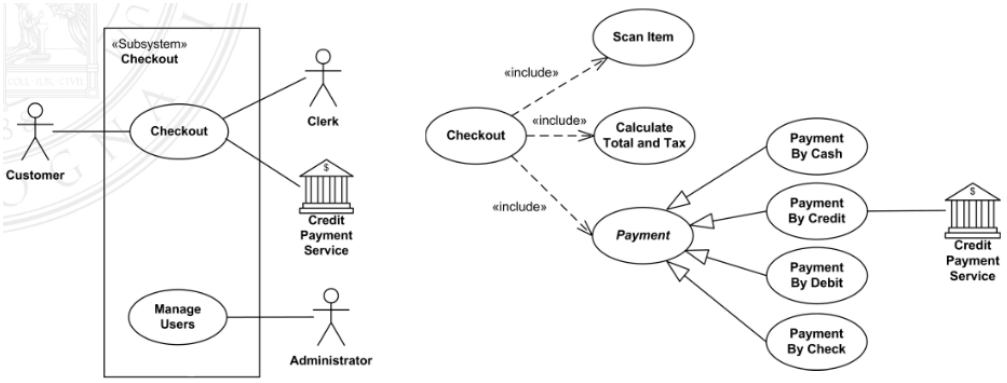
Definizione informale
Un **attore** è qualcosa caratterizzato da un comportamento, come una persona, un computer system oppure un'organizzazione.

Per cui, descrivendone maggiormente il ruolo svolto, un **attore** è un **classificatore comportamentale**, ossia un'entità che interagisce con il sistema implementato o ideato mediante **interazioni**, quali, semplicemente, l'invio di segnali o di dati.

Definzione informale
Uno **scenario**, definita anche *istanza* del caso d'uso, indica un preciso percorso attraversando certi e specifici *use case*.

Quindi il modello comportamentale trattato è definito mediante un insieme di scenari fallimentari o di successo, in relazione al conseguimento dell'obiettivo dell'attore principale, adeguandosi ai comportamenti illustrati all'interno dei singoli casi d'uso. E' bene sottolineare come ogni caso d'uso rappresenti uno specifico requisito funzionale, posto al di fuori del diagramma, dove solamente mediante interazioni è possibile stabilire un path per cui il sistema sia in grado di raggiungere l'obiettivo.
Sono definiti differenti formati del diagramma dei casi d'uso, legati ad una descrizione maggiormente esaustiva oppure funzionale; si ribadisce quale sia comunque il caposaldo su cui regge tale modello comportamentale, mantenere il *focus* sull'obiettivo principale, evitando di offuscare la mente tramite la lettura di aspetti non del tutto incidenti e pertinenti per la reale comprensione e analisi del diagramma.

Esempio
Si osserva ora una rappresentazione grafica.
Alla sinistra è posta un esempio meno dettagliato, mentre corrispettivamente è posto un livello informativo più complesso.



Non esiste un formato migliore, tuttavia la *key thing* consiste nella descrizione di dettagli dello scenario di successo e delle sue estensioni, in relazione anche a casi fallimentari, ossia quei determinati *path* che non competono per il conseguimento del *mark*, rappresentando comunque sia un *end point*.
Spesso la domanda da porsi per un qualsiasi modello adottato, consiste in cosa dovrebbe essere riportato all'interno dei differenti casi d'uso. La risposta, anche se scontata, prevede di riportare tutto ciò che permetta di soddisfare i portatori di interesse, evitando di adottare un livello specificativo troppo elevato, poichè potrebbe risultare essere fuorviante.

Modello di casi d'uso e diagramma dei casi d'uso

Un diagramma UML per un caso d'uso **NON** è un modello di caso d'uso! Il diagramma può essere visto come un **sommario**, non avendo al suo interno aspetti fondamentali come i dettagli sulle fasi di interazione e quando il caso d'uso viene applicato, le condizioni pre/post, ecc. Inoltre, **non esiste** una notazione standard per rappresentare un modello di caso d'uso, ma esistono molti **template** utilizzati in base al caso specifico. I template possono essere **molto dettagliati o più semplici**, in base ai differenti casi e dal team di sviluppo (è fondamentale rimanere costanti). In ogni template sono presenti diverse **sequenze**, che ci indicano i diversi casi possibili durante una ipotetica esecuzione. Spesso, oltre alla sequenza che illustra l'esecuzione ottimale (*Happy path*), vengono indicate sequenze alternative che portano comunque all'obiettivo finale seguendo un percorso diverso. Può anche esserci una sequenza (*Exceptions*) dove non viene raggiunto l'obiettivo.

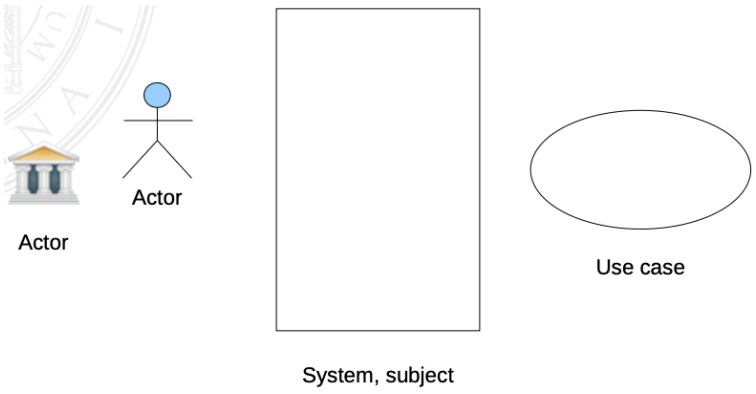
Diagramma dei casi d'uso

UML mette a disposizione il *diagramma dei casi d'uso*, illustrando i differenti *attori*, *use case* e *relazioni* che intercorrono. Le entità principali possono essere riportate come segue.

Attore
Il quale ricalca la stessa descrizione informale data precedentemente, ossia rappresenta un certo **classficatore comportamentale**.

Soggetto
Il **soggetto** è il sistema sotto analisi, al quale vengono applicati i diversi casi d'uso. Il soggetto può essere una qualsiasi entità, ossia un sistema software, sistema fisico oppure un sottosistema del soggetto. Il soggetto è un **classificatore di casi d'uso** che il ruolo di *soggetto*.

Casi d'uso
Un **caso d'uso** è anch'esso un *classificatore comportamentale* che indica quali comportamenti possano essere descritti conseguendo in una **sequenza di azioni**, pur sempre eseguite dal sistema per ottenere un risultato **osservabile**, sperando che appartenga alla zona di interesse degli stakeholder.



Relazioni

In un diagramma simile le interazioni sono espresse mediante relazioni, le quali sono di vario genere, implementate a seconda del contesto.

Relazioni attori

Generalizzazione

La **generalizzazione** tra attori è rappresentata graficamente come una linea direzionata. Adottata per esprimere l'ereditarietà, per cui da un certo attore padre posso rescindere un infinito numero di attori discendenti

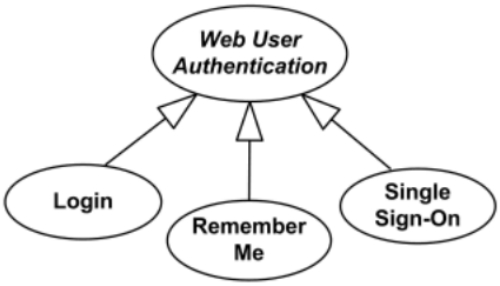
Associazione tra attori e casi d'uso

Ogni caso d'uso specifica delle funzionalità utili che il soggetto fornisce agli attori. Queste funzionalità devono essere inizializzate da un attore. L'associazione posta tra un attore e un caso d'uso può essere solo **binaria**.

Relazioni casi d'uso

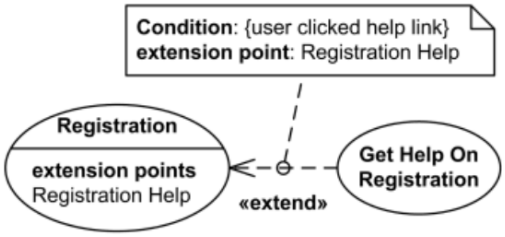
Generalizzazione

Come avviene per gli attori la **generalizzazione** tra casi d'uso avviene tramite un freccia direzionata, dove la stessa freccia è posta dal caso *specifico* a quello *generico*.



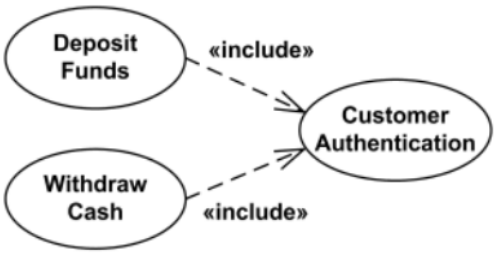
Extend

L'**extend** è una relazione che indica come e quando il comportamento definito in maniera *opzionale*, il quale estende il caso d'uso in questione, possa devenire parte del comportamento attivo. Le relazioni di questo genere vengono rappresentate graficamente come una freccia tratteggiata avente una punta aperta, in questo caso posta dal caso esteso verso il caso base; al di sopra è posta la denominazione **«extend»**.



come da rappresentazione grafica si evidenzia la presenza di un'etichetta, definita **punto di estensione**. Un punto di estensione è attuato qualora un caso d'uso specifico abbia un comportamento opzionale, ossia un'estensione rispetto al proprio *behavior standard*. Ogni punto di estensione deve avere un nome, unico e univoco nella modellazione del diagramma.

Include
Una relazione di **inclusione** è una relazione diretta tra due o più casi d'uso, adoperata quando un comportamento richiesto, quindi relativo al passaggio da requisito funzionale a behavior, non è opzionale, in cui lo stesso use case di base ne è strettamente dipendente. Essa è rappresentata mediante una freccia tratteggiata con una punta aperta, dove la direzione segue dal caso d'uso specifico verso quello base; inoltre viene posta al di sopra la denominazione **«include»**.



Main sequence

Come già detto, il diagramma dei casi d'uso non rappresenta l'unico metodo di analisi e descrizione, in relazione allo studio di modelli comportamentali, sono presenti anche approcci molto più informativi, i quali hanno un obiettivo contrario rispetto a quanto detto prima.

Un'alternativa valida è denominata **main sequence** oppure **happy path scenario**. E' in grado di descrivere il path di successo che soddisfi gli interessi degli stakeholders. Rispetto ad un qualsiasi diagramma, questa tipologia, non permette di esprimere ogni altra condizione o ramificazione. Spesso è adottata per analizzare tre punti successivi:

- Interazioni tra gli attori
- Il sistema di validazione dei comportamenti consentiti dal sistema
- Il cambiamento di stato del sistema, poichè soggetto ad influenze esterne

Tuttavia, spesso è dedicata una sezione apposita alle **estensioni**, o meglio definite **extensions**. Sono descritti gli scenari, sia di successo che fallimentari, i quali potrebbero risultare più complessi rispetto al main sequence. Gli *alternative flows* sono suddivisi in due entità

di equivalente importanza, le condizioni e le azioni.

Esempio
MAIN SUCCESS SEQUENCE (a.k.a. Happy path):

1. L'utente seleziona l'opzione per entrare in un nuovo trasferimento
2. Il sistema mostra il modulo per effettuare il trasferimento (conto destinatario, dettagli, importo, data)
3. L'utente inserisce i dati nel modulo ed effettua l'ordine
4. Il sistema mostra il messaggio di accettazione dell'ordine

EXTENSIONS:

3a Importo non disponbile:

3a.1 Il sistema mostra un messaggio di errore e torna allo step 2

Metodo di uso

I casi d'uso sono definiti per soddisfare gli obiettivi degli utenti, che interagiscono con il sistema, e soprattutto dell'attore principale. Una basica procedura può essere:

- Scegliere il sistema circondario, in relazione a quale ambiente debba essere sviluppato e modellato
- Identificare l'attore principale, solitamente posto alla sinistra, il quale compie tutte le azioni necessarie pur di conseguire nell'obiettivo voluto
- Per ogni attore identificare i loro obiettivi
- Definire i casi d'uso che consentono di soddisfare i differenti requisiti funzionali

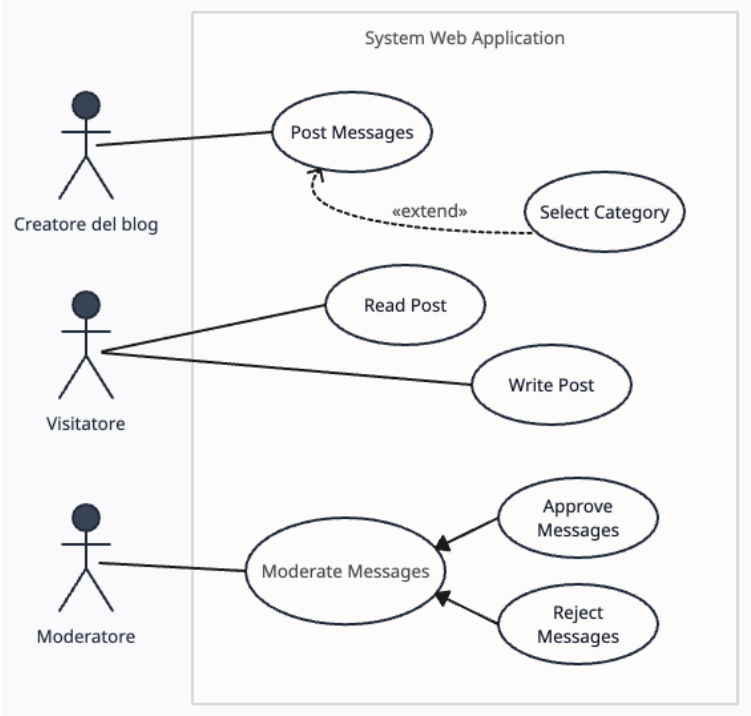
HomeWork

Un *blog* è una *web application* che contiene una serie di messaggi (*post*) datati su diversi argomenti. I messaggi vengono postati dal creatore del blog che li mette online. L'autore può associare ad ogni messaggio una o più categorie (tramite delle parole chiave).
I visitatori del blog possono commentare i messaggi; i commenti, se approvati dal moderatore (di solito è il creatore del blog), appaiono in una sezione specifica sotto il messaggio originale.

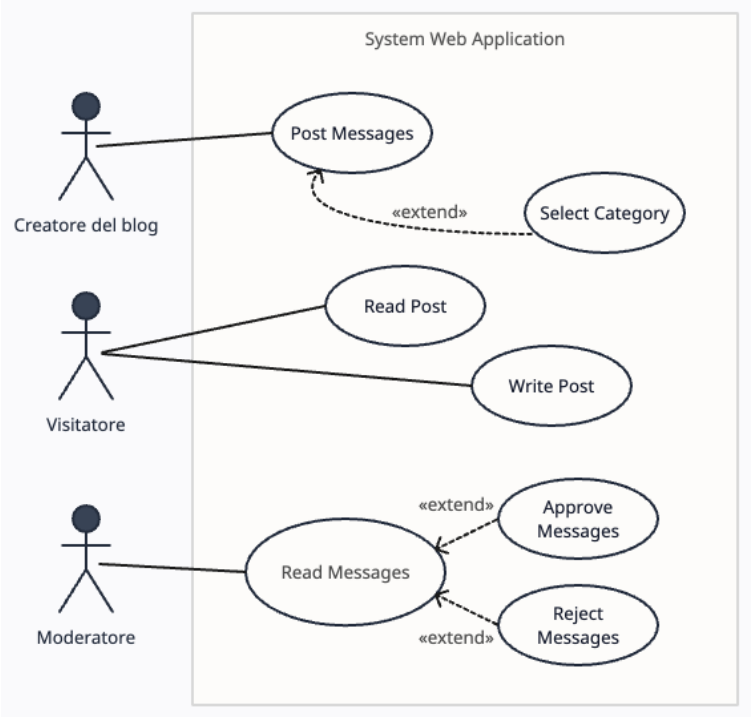
Partiamo capendo quale sia il soggetto, ovvero la nostra *web application*. Procediamo poi trovando gli attori:

- Creatore del blog
- Visitatore
- Moderatore

L'obiettivo del creatore del blog è quello di **pubblicare i messaggi**.
Vediamo ora un modello di casi d'uso d'esempio per questo caso:

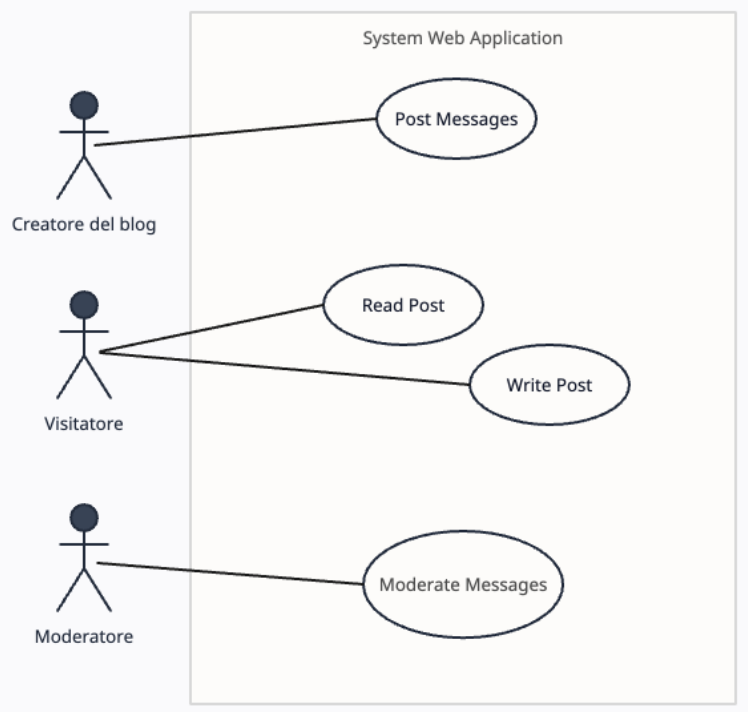


Come possiamo osservare, abbiamo i nostri tre attori (creatore del blog, visitatore e moderatore) ed i loro obiettivi. Un'altra possibile soluzione poteva essere la seguente:



In questo caso dovremmo specificare in qualche modo che non possono essere eseguite entrambe le operazioni (approve e reject), ma soltanto una per commento (magari tramite l'utilizzo di XOR). Tra i due approcci è preferibile il primo, non dovendo specificare l'esclusività e per un approccio meno confusionario rispetto al secondo. Se volessimo mantenere un mod-

ello funzionale più semplice:



Questa soluzione è buona, anche se meno dettagliata. In questo caso però il più completo è migliore, perchè non risulta essere troppo confusionario. Ripensando al ciclo di vita di un modello di casi d'uso, questa potrebbe essere la fase iniziale della modellazione.

Se avessimo voluto usare un template di caso d'uso semplice, avremmo ottenuto (stiamo considerando l'obiettivo *write comment*):

- **UC1**: write comment
- **Actors**: visitor
- **Pre-condition**: utente visualizza il messaggio
- **Main Sequence**:
  (a) utente scrive il commento
  (b) utente invia il commento
  (c) sistema conferma la ricezione del commento
- **Alternative Sequence**: il messaggio contiene linguaggio non appropriato
  (c1) Il sistema cancella il commento
- **Post-condition**: il messaggio viene aggiunto alla lista dei messaggi da moderare