

## **LINEE GUIDA - UML - CASI D'USO**

© Adriano Comai 2001-2004

Versione 2.0 del 12-3-2004

**INDICE**

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>1. PREMESSA.....</b>                                           | <b>3</b>  |
| <b>2. INTRODUZIONE.....</b>                                       | <b>4</b>  |
| 2.1 Cosa sono i casi d'uso? .....                                 | 4         |
| 2.2 Ruolo dei casi d'uso .....                                    | 4         |
| 2.3 Diagramma e modello dei casi d'uso .....                      | 4         |
| <b>3. ELEMENTI DEL MODELLO DEI CASI D'USO .....</b>               | <b>5</b>  |
| 3.1 Casi d'uso .....                                              | 5         |
| 3.2 Attori.....                                                   | 5         |
| 3.3 Sistema di riferimento (subject).....                         | 6         |
| <b>4. RELAZIONI TRA GLI ELEMENTI DEL MODELLO .....</b>            | <b>7</b>  |
| 4.1 Attori e casi d'uso .....                                     | 7         |
| 4.1.1 Attori primari e secondari.....                             | 7         |
| 4.2 Generalizzazione / specializzazione tra attori.....           | 8         |
| 4.3 Relazioni tra casi d'uso .....                                | 9         |
| 4.3.1 Generalizzazione / Specializzazione .....                   | 9         |
| 4.3.2 Include .....                                               | 10        |
| 4.3.3 Extend.....                                                 | 10        |
| 4.3.4 Considerazioni sulle relazioni tra casi d'uso .....         | 11        |
| <b>5. CASI D'USO RIFERITI A LIVELLI DIVERSI DEL SISTEMA .....</b> | <b>12</b> |
| 5.1 Strutturazione di un sistema in sottosistemi .....            | 12        |
| 5.2 Livello dei casi d'uso .....                                  | 12        |
| 5.3 Un esempio di casi d'uso a livelli diversi .....              | 13        |
| 5.4 Quanti e quali modelli definire .....                         | 15        |
| 5.5 Relazioni tra casi d'uso di livelli diversi .....             | 15        |
| <b>6. IDENTIFICARE I CASI D'USO.....</b>                          | <b>16</b> |
| 6.1 Requisiti e casi d'uso.....                                   | 16        |
| 6.2 Casi d'uso e transazioni.....                                 | 16        |
| 6.3 Casi d'uso e scenari .....                                    | 17        |
| <b>7. DOCUMENTARE I CASI D'USO .....</b>                          | <b>19</b> |
| 7.1 La documentazione testuale.....                               | 19        |
| 7.2 I diagrammi.....                                              | 19        |
| 7.2.1 Diagrammi dei casi d'uso .....                              | 19        |
| 7.2.2 Altri diagrammi UML .....                                   | 19        |
| <b>8. REALIZZARE I CASI D'USO.....</b>                            | <b>20</b> |
| <b>9. RIEPILOGO DELLE NOTAZIONI UTILIZZATE.....</b>               | <b>21</b> |

## 1. Premessa

Questa, dedicata ai casi d'uso, fa parte di una serie di linee guida per l'utilizzo di UML (Unified Modeling Language) in progetti concreti.

Le linee guida hanno un duplice obiettivo:

1. introdurre il lettore che non conosca UML al suo utilizzo. Non presuppongono conoscenze preliminari.
2. costituire, per chi abbia già una certa conoscenza del linguaggio, un riferimento per quanto riguarda la semantica e la notazione da utilizzare nei diagrammi.

## 2. Introduzione

### 2.1 Cosa sono i casi d'uso?

Ci sono modi diversi di guardare a un sistema. Uno è quello di "aprirlo" e guardarci dentro, per vedere come è strutturato all'interno. E' il punto di vista del progettista, interessato agli aspetti architettureali, ad un livello di maggiore o minore dettaglio.

Un altro punto di vista, opposto, è quello di guardare a come può essere utilizzato. In questo caso il sistema viene visto come una "black box", sigillata, ed è possibile osservarne solo i comportamenti dall'esterno. E' il punto di vista dell'utilizzatore, e di tutto ciò che interagisce con il sistema nell'ambito del suo funzionamento.

Questo secondo punto di vista corrisponde al modello dei casi d'uso. I casi d'uso (il termine "use case" è stato coniato dal metodologo svedese Ivar Jacobson) sono semplicemente i modi in cui il sistema può essere utilizzato.

Prendiamo un videoregistratore, come esempio di sistema. Possiamo aprirlo, e guardare al suo interno. Troveremo delle parti, ciascuna delle quali svolge un insieme di funzioni. E' il punto di vista del progettista, o del riparatore, ma non è ciò che fa il tipico utilizzatore del videoregistratore.

Possiamo invece descrivere come può essere utilizzato. Come si inserisce o si espelle una videocassetta, come si fa partire, come si effettua l'avanzamento veloce, il fermo immagine ... E' ciò che compare nel manuale utente del videoregistratore, e corrisponde precisamente ai casi d'uso del sistema.

### 2.2 Ruolo dei casi d'uso

I casi d'uso svolgono un duplice ruolo nello sviluppo di un sistema.

Da un lato, particolarmente nelle fasi iniziali della progettazione, servono per chiarire cosa dovrà fare il sistema. Ragionare sui casi d'uso con il committente, e con le altre funzioni interessate, è uno dei modi più efficaci ed efficienti per scoprire ed analizzare i requisiti ai quali il sistema dovrà fornire un'implementazione. Dialogare su come il sistema verrà utilizzato, nella comunicazione con persone non esperte nella progettazione di sistemi, è certamente più facile che non guardare a come dovrà essere costruito, alla sua strutturazione interna. Raggiungere un accordo con il committente sulle modalità di utilizzo del sistema consente al progettista di affrontare con maggiore tranquillità, e in separata sede, il suo mestiere specifico di progettazione.

Sull'altro fronte, i casi d'uso guidano l'intero progetto di sviluppo. Costituiscono il punto di partenza per la progettazione del sistema (lo studio dell'architettura, la creazione dei modelli di analisi e disegno, la realizzazione del codice applicativo). Sono il riferimento primario per la definizione, la progettazione, l'esecuzione dei test per la verifica di quanto prodotto. Rappresentano delle naturali unità di rilascio, per i progetti che seguono un approccio incrementale alla pianificazione della realizzazione e dei rilasci.

### 2.3 Diagramma e modello dei casi d'uso

Il modello dei casi d'uso rappresenta le funzionalità che il sistema offre ai suoi utilizzatori.

La parte veramente importante del modello è il testo, che descrive le modalità concrete con cui gli utilizzatori interagiscono con il sistema. E' il testo che descrive il caso d'uso a costituire l'elemento centrale di comunicazione tra tutte le parti in causa (committente, altre funzioni coinvolte, management, gruppo di progetto), e a guidare la progettazione e la definizione dei casi di test.

I diagrammi giocano un ruolo complementare, ma secondario. Forniscono una "mappa visuale", estremamente sintetica, degli utilizzi del sistema.

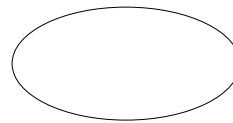
### 3. Elementi del modello dei casi d'uso

#### 3.1 Casi d'uso

Sono le funzionalità che il sistema mette a disposizione dei suoi utilizzatori.

I casi d'uso descrivono il sistema da un punto di vista esterno, come una "black box". Descrivono come verrà concretamente utilizzato. Corrispondono, in termini di contenuti, a ciò che dovrà comparire nel manuale di utilizzo del sistema (anche se il manuale utente nella sua versione definitiva è tipicamente più dettagliato di quanto non siano i casi d'uso nelle fasi iniziali del progetto, in quanto, ad esempio, riferisce gli elementi dell'interfaccia utente utilizzata per il dialogo con gli attori umani).

Nel meta-modello UML<sup>1</sup>, il caso d'uso è un tipo particolare di classificatore. Ogni istanza (specifica esecuzione) di un caso d'uso è detta scenario.



Aprire conto corrente

Il caso d'uso è rappresentato con un'icona a forma di ellisse.

#### 3.2 Attori

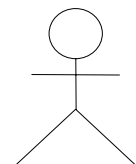
Sono i soggetti, esterni al sistema, che interagiscono con il sistema. Si può trattare di esseri umani, di organizzazioni, di altri sistemi HW o SW.

Ogni attore corrisponde ad un insieme coerente di ruoli che i soggetti esterni possono assumere interagendo con il sistema.

Gli attori interagiscono con il sistema tramite messaggi (richieste, comunicazioni, risposte).

Nel meta-modello UML, l'attore è un tipo particolare di classificatore. Poiché l'attore è, per definizione, esterno al sistema, la sua struttura interna non viene normalmente rappresentata. E' però lecito, se lo si ritiene opportuno, definire attributi ed operazioni relativi agli attori.

L'attore può avere inoltre associate una o più interfacce, che descrivano le modalità di comunicazione tra gli elementi del sistema e l'attore.



Cliente

L'attore è rappresentato con l'icona di un omino a bastoncino. E' però possibile anche definire altre icone stereotipo, per rappresentare attori non umani (cioè sistemi HW o SW esterni).

---

<sup>1</sup> Il meta-modello di UML è un modello logico, che contiene una dichiarazione del significato degli elementi UML, delle loro associazioni e delle regole per costruire modelli di sistemi che utilizzino tali elementi. In pratica, il meta-modello definisce un linguaggio per la realizzazione di modelli: UML, appunto.

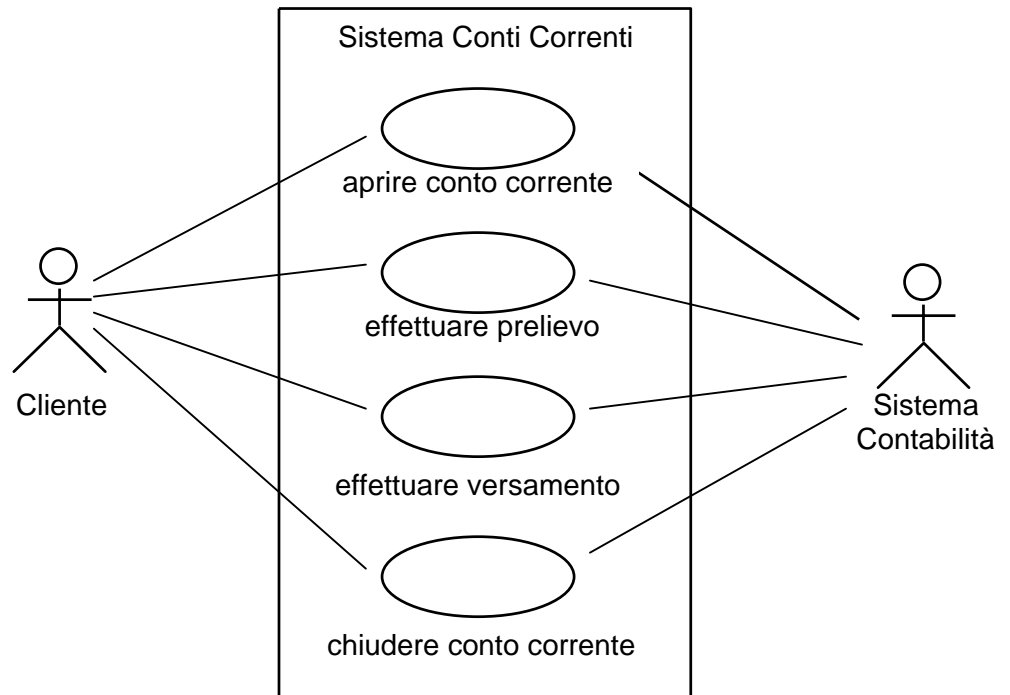
### 3.3 Sistema di riferimento (subject)

E' l'entità i cui utilizzi vengono descritti dall'insieme dei casi d'uso. Più precisamente, un insieme completo di casi d'uso descrive in modo completo gli utilizzi del sistema, ossia dal punto di vista "esterno" degli attori che interagiscono con esso, senza rivelare la struttura interna del sistema.

In UML 2.0, il sistema di riferimento per un modello di casi d'uso è chiamato "subject", e si rappresenta con un rettangolo.

Rappresenta un confine ideale, che separa ciò che è interno al sistema da ciò che ne è al di fuori. Tutti i casi d'uso rientrano all'interno del sistema di riferimento, mentre tutti gli attori sono esterni.

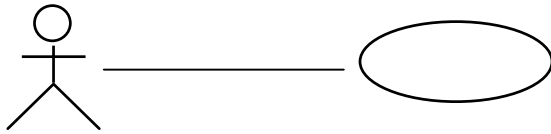
La definizione del sistema di riferimento (e quindi dei criteri di inclusione / esclusione di specifiche funzioni / attività) è una delle attività più delicate di un progetto, per la quale potrebbe essere necessario, nel corso del progetto, effettuare aggiustamenti successivi rispetto alle ipotesi iniziali.



## 4. Relazioni tra gli elementi del modello

### 4.1 Attori e casi d'uso

Ogni caso d'uso è collegato agli attori, uno o più, che partecipano al caso d'uso stesso mediante una associazione, che ha il significato di "partecipazione" (rappresentata da una linea continua che collega attore e caso d'uso).



Ogni attore può partecipare a più casi d'uso.

La comunicazione tra attori e casi d'uso, nei due sensi, è generalmente da considerarsi asincrona.

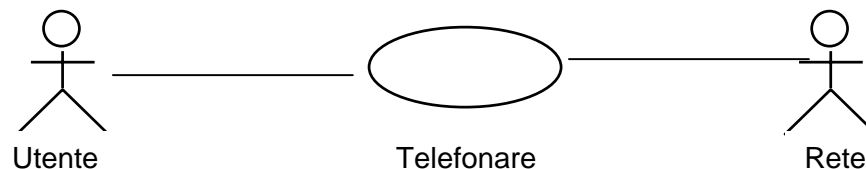
#### 4.1.1. Attori primari e secondari

La distinzione tra attori primari e secondari non è riportata nello standard UML, e non è evidenziata con notazioni diverse. Nonostante ciò, si tratta di una distinzione utile, necessaria per una corretta individuazione dei casi d'uso.

Attore primario è quello per cui è stato "pensato" il caso d'uso, in quanto ha un interesse primario per la specifica modalità di utilizzo del sistema che il caso d'uso esprime.

Attore secondario è un sistema esterno, che interagisce con il sistema di riferimento (subject) nell'ambito del caso d'uso.

Un esempio.



Il caso d'uso "Telefonare", per un sistema "telefono cellulare", ha come attore primario un "utente", e come attore secondario la "Rete".

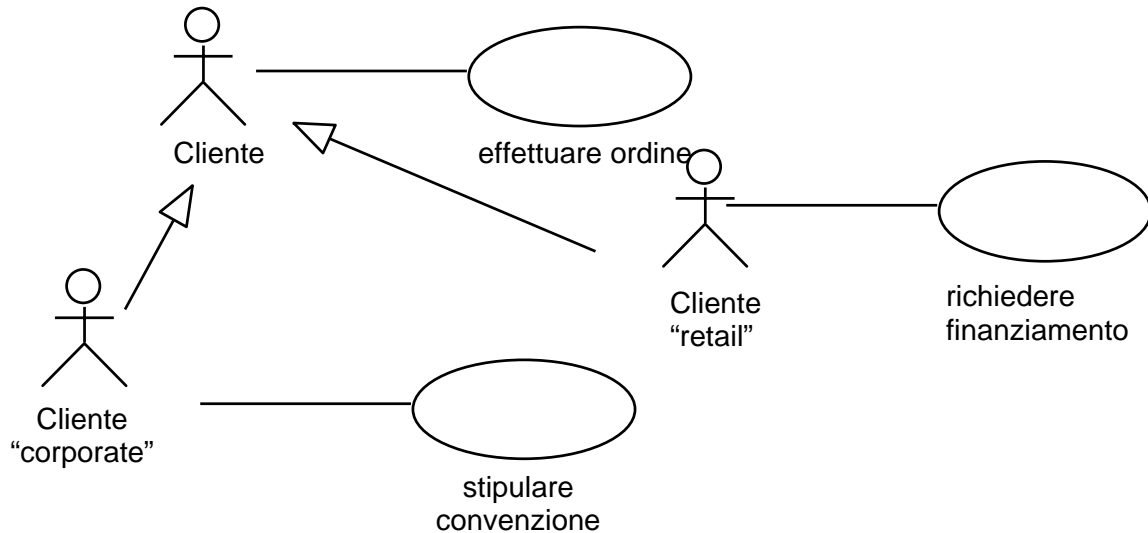
Ogni caso d'uso, per definizione, ha un attore primario. Altrimenti non sarebbe un vero caso d'uso, cioè una modalità di utilizzo del sistema che serve ad un attore (qualcuno o qualcosa esterno al sistema).

Oltre all'attore primario, ogni caso d'uso può essere collegato ad un qualsiasi numero (zero compreso) di attori secondari, in funzione della necessità del sistema di interagire con l'esterno per soddisfare l'esigenza dell'attore primario.

E' da notare che il ruolo "primario" o "secondario" non è assoluto, ma relativo ad ogni singolo caso d'uso. Un attore primario in un caso d'uso può essere secondario in un altro, e viceversa.

## 4.2 Generalizzazione / specializzazione tra attori

L'unica associazione ammessa tra attori è la specializzazione. L'attore specializzato eredita la partecipazione a tutti i casi d'uso con i quali comunica l'attore generico, ma può partecipare ad ulteriori casi d'uso ai quali l'attore generico non è collegato.



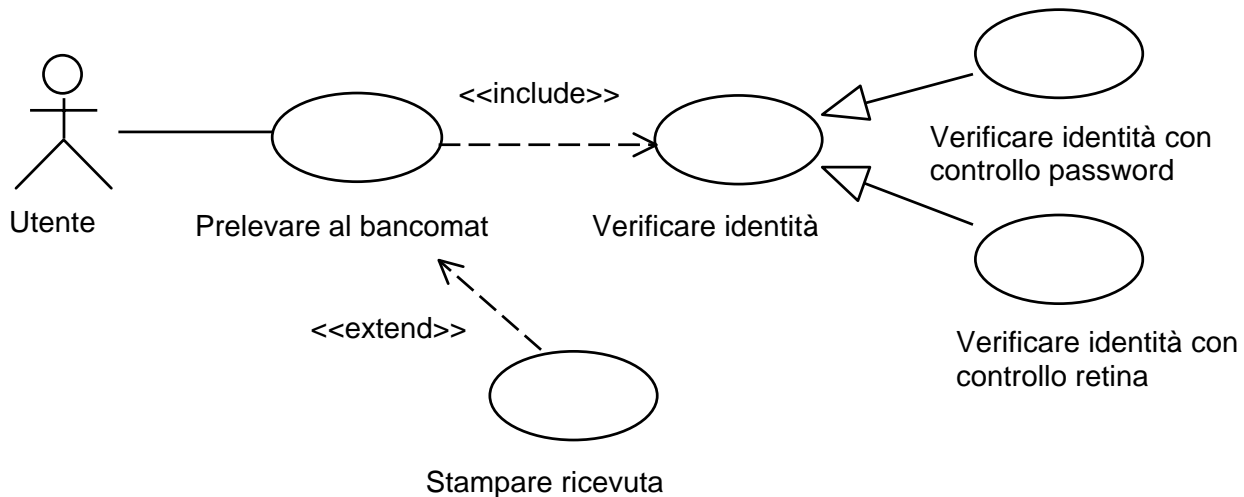
La relazione di specializzazione, in UML, è espressa graficamente con una linea continua, e con una punta di freccia triangolare e bianca.

### 4.3 Relazioni tra casi d'uso

Ogni caso d'uso descrive un utilizzo completo del sistema, e non è quindi ammessa in UML la possibilità che casi d'uso distinti abbiano tra loro un'associazione. Non è ammessa, in altri termini, la suddivisione di una funzionalità in casi d'uso separati, da eseguire in sequenza tra loro come passi di una procedura.

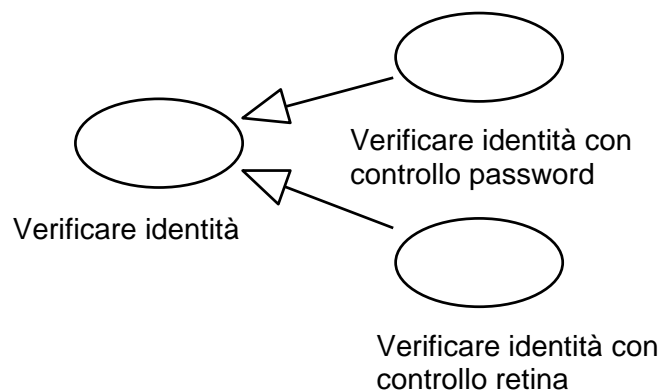
UML ammette però tre tipi di relazione:

- generalizzazione / specializzazione
- extend
- include



#### 4.3.1. Generalizzazione / Specializzazione

Associa un caso d'uso di tipo generale ad uno o più casi d'uso specializzati.



Il caso d'uso generale definisce una serie di passi, ed ha associazioni di comunicazione con uno o più attori. Ogni caso d'uso specializzato eredita le caratteristiche, i passi, gli eventuali punti di estensione (vedi *extend*) e le associazioni del caso d'uso generale.

Il caso d'uso specializzato può aggiungere nuovi passi, oppure ridefinire i passi ereditati da quello generale (override).

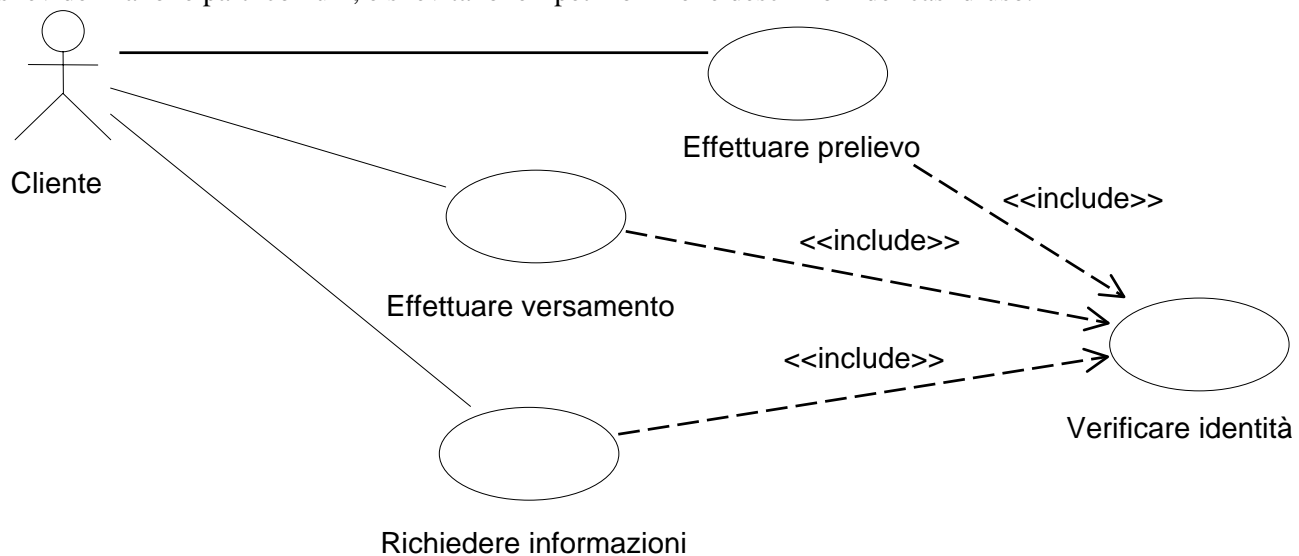
Ogni caso d'uso generale può avere più figli (casi d'uso specializzati).

Ogni caso d'uso può avere più padri, cioè specializzare diversi casi d'uso generali.

La relazione di specializzazione, come si è già detto, è espressa graficamente in UML con una linea continua, e con una punta di freccia triangolare e bianca.

### 4.3.2.Include

Casi d'uso diversi possono avere in comune una sequenza di passi da svolgere. In questo caso è possibile enucleare la sequenza comune, e definirla come un caso d'uso a sé stante, da "includere" nei casi d'uso originari. Così facendo si evidenziano le parti comuni, e si evitano le ripetizioni nelle descrizioni dei casi d'uso.



Nella sequenza di passi di ogni caso d'uso "includente" (base) viene definito un punto di inclusione per il caso d'uso "incluso". La sequenza completa dei passi del caso d'uso "includente" viene quindi a comprendere quella del caso d'uso "incluso".

Ogni caso d'uso può includere un numero illimitato di altri casi d'uso. Viceversa, ogni caso d'uso può essere incluso in un numero illimitato di altri casi d'uso.

L'associazione di inclusione è rappresentata da una dipendenza (linea tratteggiata, punta della freccia aperta) con lo stereotipo <<include>>, la cui direzione va dal caso d'uso "includente" al caso d'uso "incluso".

### 4.3.3.Extend

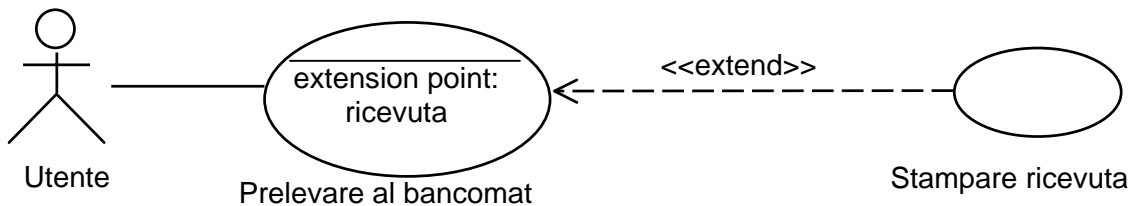
L'associazione "extend" permette di definire che un caso d'uso ("base") può venire "esteso" con il comportamento definito in un altro caso d'uso, ("di estensione"). L'estensione riguarda un comportamento opzionale del caso d'uso base, ed è soggetta ad una condizione di attivazione.

Il caso d'uso base "ignora le proprie estensioni". La sequenza dei passi che lo descrive è in sé completa, e non contiene alcun riferimento alle condizioni ed ai comportamenti definiti nel caso d'uso di estensione. L'unica particolarità che contraddistingue un caso d'uso soggetto ad estensioni è che nell'ambito della sua sequenza vengono definiti uno o più punti di estensione ("extension point"), che sono dei punti di ancoraggio ai quali si agganceranno i casi d'uso di estensione.

Ogni caso d'uso può definire più punti di estensione, ad ognuno dei quali si può agganciare più di un caso d'uso di estensione.

Il caso d'uso di estensione fa riferimento ai punti di estensione (uno o più) del caso d'uso base ai quali andrà ad agganciarsi. Nel caso i punti di estensione siano più di uno, si intende che il comportamento descritto nel caso d'uso di estensione andrà ad intercalarsi, in punti diversi, al comportamento descritto nel caso d'uso base.

Il caso d'uso di estensione, inoltre, definisce la condizione per la propria attivazione. Se la condizione viene verificata, il comportamento del caso d'uso di estensione verrà attivato nei punti di estensione opportuni; in caso contrario no.



L'associazione di estensione è rappresentata da una dipendenza (linea tratteggiata, punta della freccia aperta) con lo stereotipo <<extend>>, la cui direzione va dal caso d'uso "di estensione" al caso d'uso "base". I punti di estensione possono essere rappresentati in un comparto specifico dell'ellisse che rappresenta il caso d'uso base.

#### 4.3.4. Considerazioni sulle relazioni tra casi d'uso

Ogni relazione, come si è visto, presenta caratteristiche particolari, ed è quindi diversa dalle altre.

Nei fatti, però, è spesso possibile modellare uno specifico problema in modi diversi, utilizzando a piacere una o l'altra delle tre possibili relazioni (che non sono ortogonali, in quanto permettono di risolvere in modo diverso un medesimo problema).

⇒NOTA: Utilizzare una o l'altra delle possibili relazioni comporta come conseguenza la necessità di strutturare la descrizione testuale dei casi d'uso in modi leggermente diversi, ma non ha alcun impatto sulla struttura del sistema e sulla sua progettazione. I casi d'uso, qualunque sia la loro strutturazione, non definiscono né rivelano in alcun caso la struttura del sistema.

L'effetto globale dell'utilizzo delle relazioni tra casi d'uso è quello di una frammentazione del singolo caso d'uso, anche se basata sulla "emersione" di specializzazioni, o di particolarità opzionali (extend) o di comportamenti comuni (include), anziché sulla suddivisione di una funzionalità completa in una sequenza di passi o di parti, come avviene nella scomposizione funzionale top-down tipica di un approccio strutturato.

Nell'esempio riportato in figura, un singolo caso d'uso originario, "Prelevare al bancomat", è stato scomposto grazie alla definizione degli altri casi d'uso che sono associati ad esso, "Stampare ricevuta" e "Verificare identità" (quest'ultimo è stato ulteriormente scomposto, dando origine ai casi d'uso che lo specializzano).

⇒NOTA: La maggioranza degli esperti sconsiglia l'utilizzo di questi meccanismi di relazione. La proliferazione dei casi d'uso che può scaturirne può produrre un modello parcellizzato, frammentato, difficile da gestire.

Si consiglia quindi di limitare l'utilizzo dei meccanismi di relazione dei casi d'uso, e di ricorrervi solo quando il nuovo caso d'uso, generato dalla scomposizione, abbia una rilevanza tale da costituire un elemento significativo per il dialogo tra le diverse funzioni coinvolte nel progetto, ed in particolare nelle comunicazioni tra committenti e fornitori.

## 5. Casi d'uso riferiti a livelli diversi del sistema

### 5.1 Strutturazione di un sistema in sottosistemi

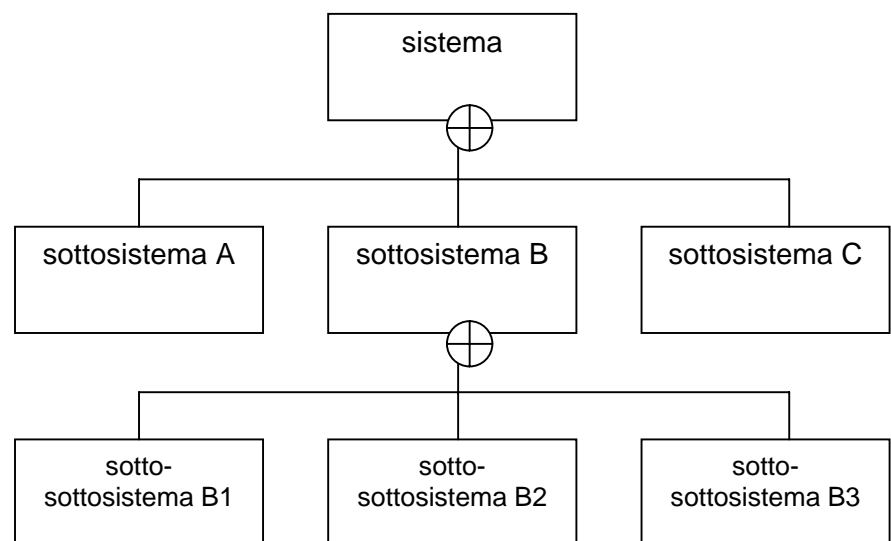
Ogni sistema può essere strutturato in parti distinte, che interagiscono tra loro per fornire le funzionalità complessive del sistema stesso.

È così possibile definire un sistema a livello organizzativo, dove la parte "sistema software" costituisce solo una delle componenti del sistema complessivo. Ad esempio, è possibile rappresentare un sistema di sportello bancario come un sistema complesso: alcune funzioni del sistema verranno svolte da esseri umani, altre da un sistema informatico.

Per i sistemi embedded, in cui la componente software viene eseguita su hardware dedicato, è quasi naturale rappresentare il sistema su più livelli: il sistema complessivo (hardware più software); il (la parte di) sistema hardware; il (la parte di) sistema software.

Focalizzandosi sull'ambito specifico dei sistemi software, ogni sistema non banale può essere articolato in sottosistemi distinti, che interagendo tra loro contribuiranno a fornire le funzionalità complessive del sistema globale.

Tale articolazione / scomposizione può essere effettuata in modo ricorsivo, suddividendo ogni parte del sistema in parti sempre più piccole. In UML l'elemento finale, non ulteriormente scomponibile, è la classe.



Nella figura, la strutturazione gerarchica del sistema è rappresentata utilizzando la notazione UML che rappresenta un "contenimento", cioè un cerchio bianco che ha al suo interno un +.

### 5.2 Livello dei casi d'uso

I casi d'uso possono descrivere gli utilizzi di un sistema - a qualsiasi livello, sia organizzativo che informatico. Immaginiamo un sistema organizzativo, che abbia come una propria parte un sottosistema software. In questa situazione è possibile definire almeno due insiemi distinti di casi d'uso:

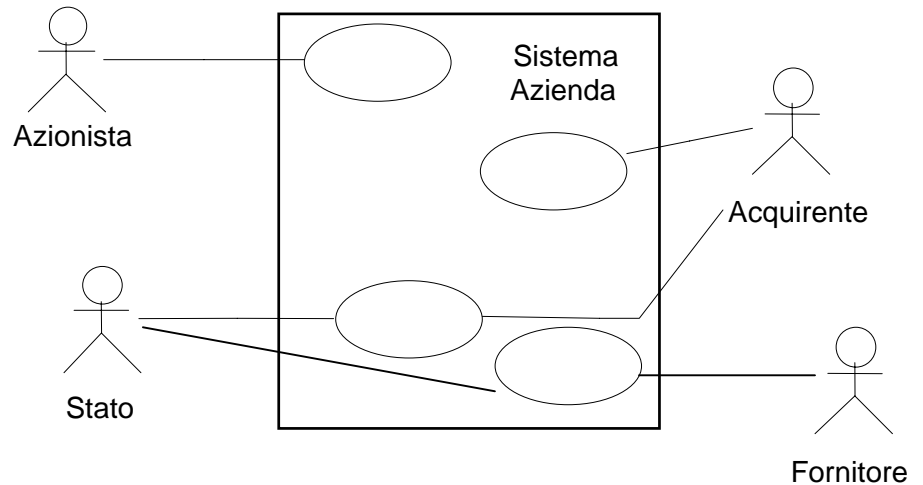
- uno per il sistema globale (che comprende aspetti automatizzati e non)
- uno per il sistema software (che è una parte specifica del sistema globale)

Ad ogni livello in cui il sistema complessivo può essere strutturato, partendo dal livello globale, può corrispondere la definizione di un insieme di casi d'uso. Comunque definiti, i casi d'uso che descrivono un certo livello ne specificano le modalità di utilizzo, ma non la strutturazione interna, ossia l'articolazione di quel livello in sottosistemi.

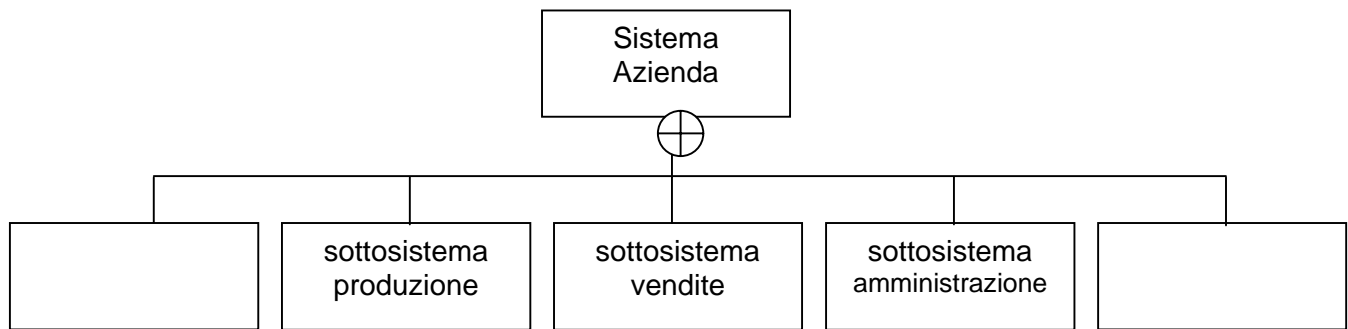
**Per ogni insieme di casi d'uso che specifica un sistema (cioè un particolare livello) deve essere definito uno specifico contesto (subject). Ogni livello di sistema, infatti, ha un proprio contesto, ed un proprio insieme di attori con il quale interagisce.**

### 5.3 Un esempio di casi d'uso a livelli diversi

E' possibile identificare e descrivere i casi d'uso per un sistema che coincide con un'intera azienda ("Sistema Azienda"). Gli attori saranno tutti i soggetti esterni che interagiscono con l'azienda (es. clienti, fornitori, azionisti). I casi d'uso saranno tutte le modalità di "utilizzo" dell'azienda significative per gli attori.



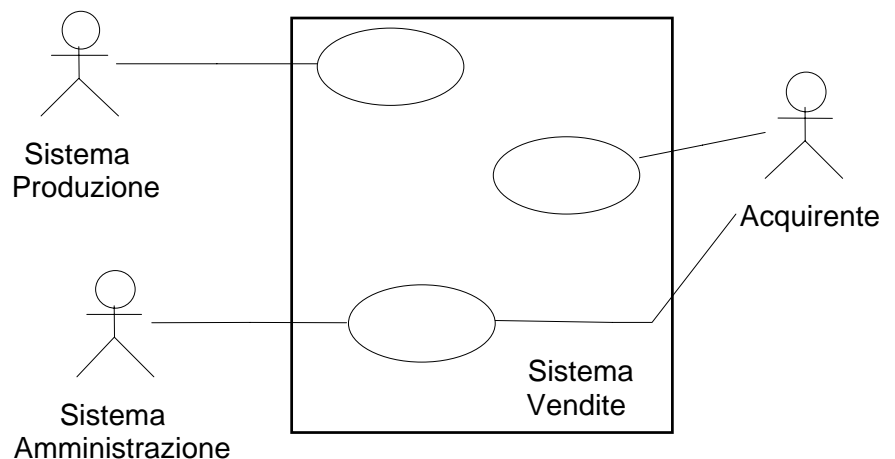
Il "Sistema Azienda", naturalmente, può essere articolato in sottosistemi.



Per ognuno dei sottosistemi individuati può essere definito un modello dei casi d'uso specifico.

Ogni sottosistema corrisponderà ad un contesto specifico (subject), ed avrà propri attori e propri casi d'uso. Alcuni attori sono esterni all'azienda (e comparivano quindi già nel contesto del "Sistema Azienda"); altri invece coincidono con gli altri sottosistemi, che interagiscono con quello sotto osservazione.

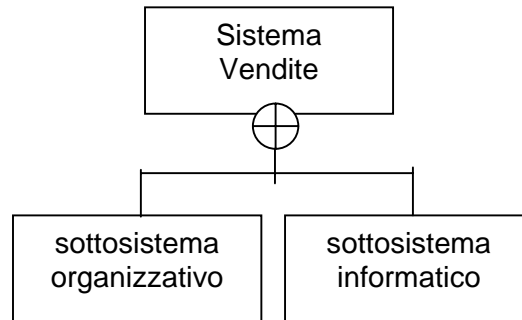
Nel nostro esempio, il "Sistema vendite" ha come attori l'acquirente, che compariva già come attore nel contesto del "Sistema Azienda" (il livello superiore), ed i sottosistemi "Produzione" e "Amministrazione", che costituiscono invece altre parti del "Sistema Azienda".



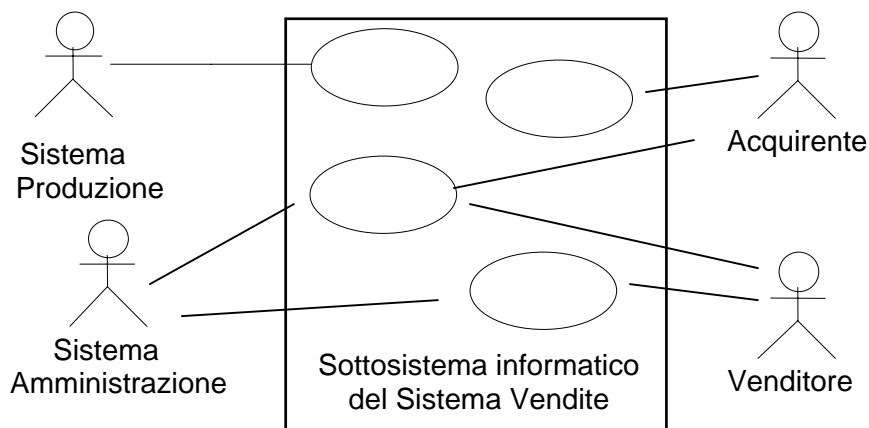
Se analizziamo a questo punto la struttura interna del "Sistema Vendite", scopriamo che comprende alcuni ruoli e attività svolte da funzioni organizzative, cioè da esseri umani, descrivibili, se il caso, in termini di procedure e mansionari.

Altre attività vengono invece svolte da uno o più sistemi informatici.

In forma semplificata, possiamo rappresentare un'articolazione strutturale del sistema vendite in due sottosistemi, uno "organizzativo", l'altro "informatico".



Scendendo di livello, il contesto del "sottosistema informatico del sistema vendite" avrà come attori alcuni tra gli attori che comparivano nel livello superiore, ossia il "Sistema Vendite"; altri attori, nello specifico il "Venditore", corrisponderanno invece ai ruoli svolti dalle funzioni organizzative interne al "sottosistema organizzativo", cioè da esseri umani.



L'insieme dei casi d'uso che descrivono gli utilizzi di questo livello, ossia del "Sottosistema informatico del sistema vendite", definisce ciò che dovrà essere realizzato da un progetto informatico. Gli attori che compaiono a questo livello saranno gli utilizzatori del sistema informatico, o altri sistemi che dovranno interoperare con tale sistema.

Per concludere l'esempio, anche il sottosistema informatico del sistema vendite può essere a sua volta strutturato in sottosistemi, per ognuno dei quali è definibile un contesto ed un insieme di casi d'uso, e così via. Come si è detto, il punto terminale teorico di questo processo di scomposizioni successive è, in UML, la classe.

E per ogni classe, che può essere vista a tutti gli effetti come un sistema di dimensioni estremamente ridotte, è possibile definire un insieme di casi d'uso, che corrisponderanno tipicamente alle operazioni pubbliche, e quindi ai servizi che possono essere richiesti alla classe ed ai suoi oggetti. Gli attori, in questo contesto, saranno le altre classi che possono richiedere servizi alla classe ed ai suoi oggetti, o alle quali la classe a sua volta richiede servizi. La definizione di un modello dei casi d'uso per ogni singola classe è una possibilità esplicitamente contemplata da UML, anche se dal punto di vista pratico risulta raramente utile.

## 5.4 Quanti e quali modelli definire

Nei casi in cui può essere utile un'analisi del flusso del processo organizzativo, per chiarire ruoli e responsabilità, può essere definito un modello dei casi d'uso a livello "business", che evidenzia il contesto organizzativo in cui il sistema informatico andrà a collocarsi. Nel modello dei casi d'uso business, il sistema complessivo comprende sia il sistema informatico che ruoli e attività svolte da esseri umani.

Analogamente, nel caso di sistemi embedded, può essere definito un modello dei casi d'uso a livello di sistema complessivo, che comprende sia i componenti hardware che quelli software.

Nei casi in cui sia necessario realizzare un sistema software, comunque, è opportuno definire un modello dei casi d'uso ad un livello che abbia come contesto (cioè si riferisca ad un sistema – subject – che comprende al suo interno) tutte le funzionalità software da realizzare, e come attori gli utilizzatori umani del sistema e gli altri sistemi, hardware o software, con cui il sistema software dovrà interagire.

## 5.5 Relazioni tra casi d'uso di livelli diversi

Ogni caso d'uso, come si è detto, è riferito ad uno specifico livello di sistema, e quindi ad uno specifico contesto (subject), del quale descrive una specifica modalità di utilizzo in modo completo.

Quando si hanno modelli diversi di casi d'uso, ogni modello è riferito ad un diverso livello di sistema.

E' normale che un caso d'uso, riferito ad uno specifico livello di sistema, corrisponda ad uno o più casi d'uso riferiti ad un livello di sistema inferiore nella gerarchia.

Nel nostro esempio, a livello di modello dei casi d'uso del "Sistema Vendite", l'attore "Acquirente" partecipa ad un caso d'uso "Acquistare prodotto".

Ad un livello inferiore, nel modello dei casi d'uso del "Sottosistema informatico del sistema vendite", l'attore "Venditore" partecipa ad un caso d'uso "Acquisire ordine cliente", un attore "Magazziniere" partecipa ad un altro caso d'uso "Impegnare prodotti su ordine", un altro attore "Spedizionario" partecipa ad un caso d'uso "Consegnare prodotti ordinati".

La relazione tra il caso d'uso di livello superiore, che in UML è detto "superordinato", e quelli di livello inferiore, detti "subordinati", è una relazione di "raffinamento" ("refine"). Il comportamento descritto da un caso d'uso superordinato viene così "raffinato" dal comportamento di un insieme di casi d'uso subordinati, ciascuno dei quali è associato ad attori specifici, che al livello superiore potevano essere nascosti, in quanto facenti parte dei confini del sistema.

Il fatto che la relazione tra casi d'uso superordinati e subordinati sia di raffinamento, e non di composizione, è significativo, perché preclude la possibilità di una scomposizione funzionale dei casi d'uso. In nessun caso si può dire che un caso d'uso è "composto" da altri casi d'uso.

## 6. Identificare i casi d'uso

Nel caso si vogliano identificare i casi d'uso per un sistema già esistente, il punto di partenza migliore è sicuramente il manuale di utilizzo, se disponibile e aggiornato, cioè conforme a quanto il sistema effettivamente fa. In alternativa, i casi d'uso di un sistema esistente possono essere identificati intervistando gli utilizzatori, o attraverso l'utilizzo diretto del sistema stesso.

Per un sistema nuovo, o per un progetto di evoluzione di un sistema esistente, si può seguire la traccia seguente.

1. Partire dai requisiti già rilevati, attraverso il dialogo con il committente e le altre funzioni interessate, e/o attraverso l'esame della documentazione iniziale disponibile per il progetto.
2. Identificare tutte le tipologie di utilizzatori del sistema (esseri umani o altri sistemi). Costituiranno, almeno in prima battuta, attori primari per i casi d'uso da individuare nel contesto del sistema.
3. Per ogni attore primario, rilevare in quale modo può / deve utilizzare il sistema, partendo dagli obiettivi che deve raggiungere. Ad ogni modalità di utilizzo corrisponde, almeno in prima battuta, un caso d'uso.
4. Per ogni caso d'uso, descrivere lo scenario base (la sequenza di passi più semplice possibile che conduce al successo del caso d'uso, le risposte attese dal sistema), e le principali varianti a tale scenario. Così facendo, tipicamente, possono emergere necessità di interazione del sistema con altri soggetti (esseri umani o altri sistemi), che verranno rappresentati nel modello come attori secondari.

### 6.1 Requisiti e casi d'uso

Un caso d'uso può essere associato a più requisiti funzionali e non funzionali; un requisito funzionale può dare origine a più casi d'uso.

I requisiti, cioè ciò che viene richiesto dal committente e dalle altre funzioni interessate in merito alle caratteristiche del sistema, costituiscono il punto di partenza per l'individuazione dei casi d'uso. Nello stesso tempo, i casi d'uso costituiscono uno strumento estremamente efficace ed efficiente per portare alla luce nuovi requisiti non ancora emersi, e per chiarificare requisiti ambigui, generici o in conflitto tra loro.

L'efficacia dei casi d'uso per la scoperta dei requisiti è legata al fatto che, nel dialogo tra chi progetta e le altre funzioni interessate, vengono analizzati per ogni caso d'uso gli scenari concreti di operatività degli utilizzatori nei confronti del sistema, le sequenze di passi, le varianti e le eccezioni. In questo modo il committente e le altre funzioni interessate possono esprimere feedback sugli scenari concreti ipotizzati da chi progetta, feedback che chiariscono requisiti esistenti o ne specificano di nuovi.

La conclusione del dialogo porta ad un chiarimento degli scenari, ad una versione concordata del caso d'uso, e a un elenco di requisiti più dettagliato e con minori ambiguità. Ma l'aspetto più importante è che i casi d'uso concordati costituiscono, a tutti gli effetti, dei requisiti concordati per il sistema da sviluppare, ed il riferimento principale per la definizione, progettazione ed esecuzione dei test da effettuare per la verifica del sistema in sede di accettazione.

### 6.2 Casi d'uso e transazioni

Operativamente, un caso d'uso può corrispondere a più transazioni informatiche che il sistema deve eseguire per produrre il comportamento ed i risultati attesi.

Il caso d'uso viene definito a partire da una visione esterna del sistema: quella di uno specifico attore primario, per il quale costituisce una particolare modalità di utilizzo. La definizione del caso d'uso, da questo punto di vista esterno, deve essere completa, deve, cioè, fornire tutte le risposte legate all'evento che fa iniziare il caso d'uso stesso, indipendentemente dal fatto che il sistema, per fornire tali risposte, debba svolgere una o più transazioni, magari in momenti diversi.

Viceversa, una transazione informatica ha caratteristiche precise (atomicità, isolamento rispetto alle altre transazioni, consistenza e effetto durevole nel tempo degli aggiornamenti effettuati), che corrispondono ad un'azione generalmente più limitata rispetto a quella attribuibile ad un caso d'uso.

Ad esempio, in un sistema di commercio elettronico, "Acquistare un prodotto" via internet è un caso d'uso per l'attore "Cliente", che si conclude, in caso di successo, con l'arrivo a casa della merce ordinata, l'addebito dell'importo corretto sulla carta di credito ecc.

Per raggiungere tale risultato, al sistema sarà necessario effettuare certamente più transazioni, in tempi diversi, con il coinvolgimento probabile di diverse funzioni organizzative interne. Sarebbe però sbagliato definire ciascuna di queste transazioni come un caso d'uso a sé stante, almeno al livello ("business") in cui questo particolare utilizzo del sistema (caso d'uso) viene descritto dal punto di vista del cliente, ed il sistema stesso (composto di software ma anche di attività svolte da esseri umani, quindi organizzativo) deve essere considerato come una black-box. Se si definisse un caso d'uso per ognuna delle transazioni interne al sistema, infatti, si perderebbe la visione complessiva delle azioni da effettuare per fornire le risposte necessarie alla richiesta dell'attore.

A livello di sistema informatico, invece, è più probabile che un caso d'uso corrisponda ad una singola transazione. Ciò nonostante, ogni caso d'uso va definito dal punto di vista esterno dell'attore primario, cui deve fornire risposte complete, e può quindi corrispondere a più transazioni informatiche dal punto di vista implementativo.

### 6.3 Casi d'uso e scenari

In UML, ogni specifica esecuzione (istanza) di un caso d'uso è detta scenario. Ad esempio, in un caso d'uso "Acquistare un prodotto", ogni specifico acquisto effettuato da uno specifico cliente in uno specifico momento costituisce uno scenario particolare.

Ogni scenario, pur restando nell'ambito generale della modalità di utilizzo del sistema definita dal caso d'uso, presenta caratteristiche particolari.

Innanzitutto, esistono scenari di successo (nel nostro esempio, l'acquisto va a buon fine) e scenari di fallimento (tutti quelli in cui, per i motivi più vari, l'acquisto non va a buon fine). Ma più in generale, le strade che possono condurre al successo o al fallimento di un caso d'uso sono numerose, e gli scenari possibili sono illimitati.

Ogni scenario può essere descritto in una sequenza di passi, che specificano l'interazione (il dialogo, fatto di stimoli e risposte) che avviene tra il sistema e gli attori coinvolti nello scenario.

Ad esempio:

- 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

Ogni passo dello scenario descritto, ovviamente, può essere soggetto a condizioni, eccezioni, varianti, ciascuna delle quali può a sua volta condurre ad un nuovo scenario.

In termini di descrizione, la prassi più diffusa per la specifica degli scenari di un caso d'uso è quella di definire uno scenario base, cioè lo scenario più semplice possibile che porta al successo del caso d'uso. Allo scenario base, o per essere più precisi ai singoli passi che lo specificano, vengono quindi "agganciate" le varianti, ciascuna delle quali costituisce una diramazione che può portare al successo o al fallimento del caso d'uso.

Ad esempio, per un caso d'uso "Aprire conto corrente bancario":

Scenario base:

- 1 il cliente si presenta in banca per aprire un nuovo c/c
- 2 l'addetto riceve il cliente e fornisce spiegazioni
- 3 il cliente accetta, e fornisce i propri dati
- 4 l'addetto verifica se il cliente è censito in anagrafica
- 5 l'addetto crea il nuovo conto corrente
- 6 l'addetto segnala il numero di conto al cliente

Varianti:

- 3(a) se il cliente non accetta il caso d'uso termina

3(b) se il conto va intestato a più persone vanno forniti i dati di tutte

4(a) se il cliente (o uno dei diversi intestatari) non è censito l'addetto provvede a registrarlo, richiede al cliente la firma dello specimen e ne effettua la memorizzazione via scanner

La distinzione tra caso d'uso e i suoi scenari, che a livello teorico può sembrare netta, all'atto pratico può risultare invece molto più sfumata.

In un sistema di vendite online, ad esempio, è possibile definire i tre casi d'uso "Introdurre un nuovo prodotto a catalogo", "Aggiornare caratteristiche di un prodotto esistente", "Eliminare un prodotto dal catalogo", ognuno dei quali può essere descritto con uno scenario base ed una serie di varianti, come indicato in precedenza.

E' anche possibile e lecito, però, definire un unico caso d'uso, "aggiornare catalogo prodotti", che comprenda le tre diverse modalità di utilizzo. In questa seconda ipotesi, le tre modalità di utilizzo possono essere descritte come tre scenari dello stesso caso d'uso, alternativi tra loro, ciascuno corredato dalle rispettive varianti.

La sostanza di ciò che il sistema deve fare non cambia, qualunque sia la soluzione scelta, ed anche a livello di specifica le differenze sono minimali. Ma ciò che nella soluzione "più frammentata" costituiva un caso d'uso a sé stante diventa, nella soluzione "più aggregata" un semplice scenario.

## 7. Documentare i casi d'uso

### 7.1 La documentazione testuale

Né Ivar Jacobson (che li ha inventati) né UML, che li ha resi standard, hanno specificato il modo in cui va descritto un caso d'uso, lasciando spazio libero a tutte le possibili forme di documentazione testuale.

L'unico dato certo, quindi, è che la descrizione testuale del caso d'uso, comunque effettuata, ha un ruolo centrale nella comunicazione tra i diversi soggetti coinvolti nello sviluppo di un sistema, dal committente agli utilizzatori, da chi progetta agli specialisti di test.

Per un esempio di documentazione testuale, si veda la template di descrizione sul sito [www.analisi-disegno.com](http://www.analisi-disegno.com).

### 7.2 I diagrammi

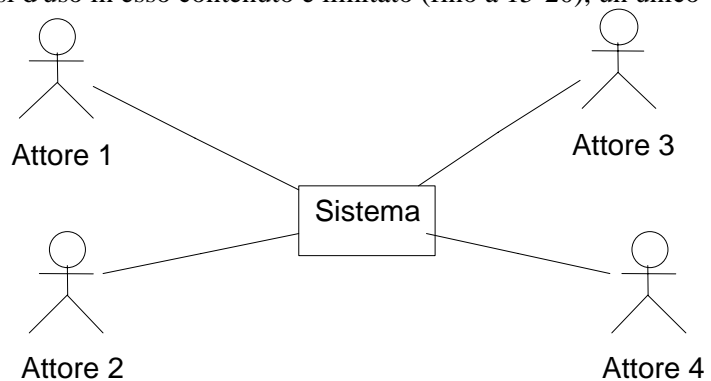
#### 7.2.1. Diagrammi dei casi d'uso

I diagrammi dei casi d'uso forniscono una mappa visuale degli utilizzi del sistema. Per ogni modello dei casi d'uso, riferito ad uno specifico livello di sistema di riferimento, è possibile definire uno o più diagrammi.

Se il modello è semplice, cioè se il numero dei casi d'uso in esso contenuto è limitato (fino a 15-20), un unico diagramma può essere sufficiente.

Quando invece il numero dei casi d'uso è elevato, un unico diagramma può risultare poco comprensibile.

Una possibile soluzione è quella di definire un diagramma delle classi estremamente sintetico, che comprenda tutti gli attori e evidenzi la loro associazione nei confronti del sistema globale, rappresentato come un'unica classe.



A questo primo diagramma, che rappresenta in forma sintetica il sistema nella globalità delle sue interazioni con l'ambiente esterno, si possono quindi aggiungere una serie di diagrammi dei casi d'uso, parziali, che rappresentano le modalità di utilizzo del sistema da parte di un singolo attore.

Ogni diagramma riporterà solo i casi d'uso a cui partecipa uno specifico attore (anche se in esso compariranno anche gli altri attori associati a tali casi d'uso), e fornirà quindi una visione complessiva degli utilizzi del sistema dal punto di vista di quell'attore.

#### 7.2.2. Altri diagrammi UML

Tutti gli altri diagrammi UML sono in qualche modo legati ai casi d'uso, dal momento che i casi d'uso guidano l'intera attività di progettazione del sistema.

In particolare, il diagramma delle classi ed i diagrammi di interazione vengono usati per modellare la realizzazione dei casi d'uso, mentre il diagramma dei componenti e quello di deployment (distribuzione) permettono di specificare l'architettura di sistema che dovrà implementare i casi d'uso.

Un ruolo specifico può però essere svolto dai diagrammi di stato e di attività.

È possibile utilizzare un diagramma di stato per rappresentare l'evoluzione degli stati, cioè delle condizioni in cui il sistema si può trovare durante l'esecuzione del caso d'uso.

Ed è possibile rappresentare, con un diagramma di attività, la sequenza dei passi e le condizioni che specificano uno o più scenari del caso d'uso.

## 8. Realizzare i casi d'uso

La realizzazione dei casi d'uso è influenzata in generale da due fattori: la tipologia del sistema di cui descrivono l'utilizzo, e l'approccio utilizzato per la progettazione.

Se il sistema a cui il caso d'uso si riferisce è un sistema di "business" composto sia da parti organizzative che da parti informatiche, la realizzazione del caso d'uso consisterà nella individuazione delle responsabilità da attribuire ai singoli ruoli organizzativi, di quelle da attribuire al sistema informatico, e delle modalità di interazione che ogni specifico ruolo dovrà avere nei confronti degli altri ruoli organizzativi e del sistema informatico.

La realizzazione dei casi d'uso business rientra tra le attività della disciplina di Business Modeling.

Se il sistema a cui il caso d'uso si riferisce è un sistema software, la realizzazione del caso d'uso dipende dall'approccio utilizzato per la progettazione, che può essere object oriented oppure strutturato - funzionale.

In una progettazione object oriented, la realizzazione del caso d'uso comporta l'individuazione delle classi che vi parteciperanno e delle rispettive responsabilità, nonché la definizione delle modalità di interazione tra gli oggetti delle classi coinvolte.

In una progettazione strutturata, invece, si tratterà di definire le procedure e le transazioni coinvolte, e la struttura gerarchica dei moduli software.

La realizzazione dei casi d'uso di un sistema software rientra tra le attività della disciplina di analisi e design.

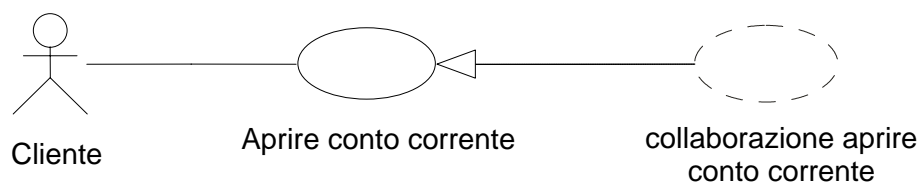
Se, infine, il sistema a cui il caso d'uso si riferisce è una singola classe, la realizzazione del caso d'uso consisterà nell'implementazione delle operazioni che devono soddisfare il caso d'uso, cioè nella scrittura dei metodi in uno specifico linguaggio di programmazione.

In UML, la realizzazione dei casi d'uso può essere espressa con una "collaborazione". La collaborazione è costituita da una serie di oggetti, che interagendo tra loro (effettuando delle azioni e scambiandosi dei messaggi, per richiedere servizi corrispondenti alle rispettive responsabilità) svolgono i passi specificati nel caso d'uso.

La collaborazione che realizza un caso d'uso (o uno scenario di caso d'uso) può essere descritta:

- a livello statico mediante un diagramma delle classi, che evidenzia le classi o gli oggetti coinvolti nella collaborazione, e le rispettive associazioni e legami
- a livello dinamico mediante uno dei diagrammi di interazione (ad esempio, sequenza), che evidenzia i messaggi che gli oggetti si scambiano nell'ambito della collaborazione.

L'icona che rappresenta una collaborazione in UML è una ellisse tratteggiata. La relazione tra un caso d'uso e la collaborazione che lo realizza è, appunto, una relazione di realizzazione, rappresentata con una linea tratteggiata con punta della freccia triangolare e bianca.



9. Riepilogo delle notazioni utilizzate

