
Withdraw money from an ATM

Primary actors: Customer
ATM Technician
Bank

Preconditions: Network connection is active
ATM has available cash

Basic flow of events:

1. Bank customer inserts debit card and enters PIN.
2. Customer is validated.
3. ATM displays actions available on ATM unit. Customer selects Withdraw Cash.
4. ATM prompts account.
5. Customer selects account.
6. ATM prompts amount.
7. Customer enters desired amount.
8. Information sent to Bank, inquiring if sufficient funds/allowable withdrawal limit.
9. Money is dispensed and receipt prints.

Alternative flows

- 2a. Customer is not validated.
 - 2a1. ATM displays error message.
- 7a. Customer selects invalid amount.
 - 7a1. ATM prompts user to re-enter valid amount.
- 8a. Customer has insufficient funds.
 - 8a1. ATM displays error message.
 - 8a2. ATM shows available withdrawal limit, redirects to step 6.
- 9a. ATM has insufficient cash.
 - 9a1. ATM Technician is alerted.
 - 9a2. ATM displays error message and phone number to call.
 - 9b. Cash gets stuck in dispensing.
 - 9b1. ATM displays error message.

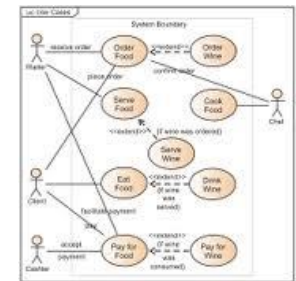


DEFINIZIONE DEI REQUISITI BASATA SU USE CASE

Ingegneria del Software 2023-2024

AGENDA (LEZIONE DI OGGI)

- Cosa sono gli **Use Case**?
- Per cosa sono usati?
- **Elementi della definizione** dei requisiti basata su use case
 - Attori, Use case e relazioni
- **Use Case Diagram**
- Template disponibili per gli Use case
 - Ne vedremo uno ...
- Consigli su come scriverli
- Screen Mockup e GUI prototyping



A screen mockup of an ATM interface. The window title is 'ATM'. The text inside says 'Welcome!' followed by 'Please enter your account number:' and 'Enter your PIN:'. Below this is a text input field containing '54321'. At the bottom, there is a numeric keypad with buttons 1-9, 0, C, and ENTER. To the right of the keypad are two labels: 'Withdraw' and 'Insert Money', each followed by a dashed line for input.

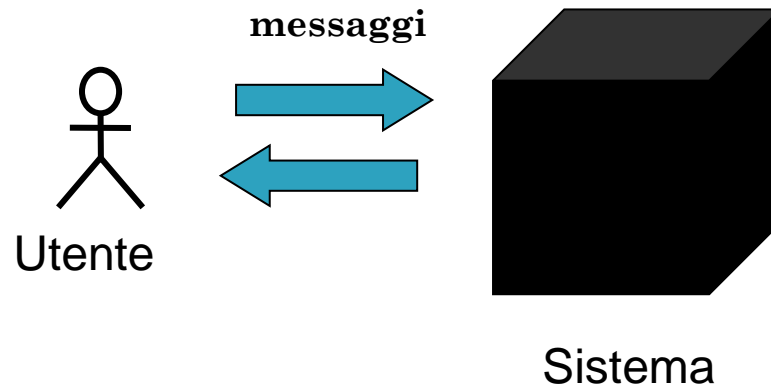
USE CASE (CASI D'USO)



- Proposti da **Ivar Jacobson** nel 1992
- Tecnica per:
 - **esprimere requisiti funzionali** di un sistema
 - descrivere dal punto di vista di chi lo usa (senza guardare come è fatto) un sistema
 - sistemi di vario genere, es.
 - organizzativo complesso (es. porto o banca)
 - quello software/hardware da sviluppare
 - anche una componente/un sottosistema
- **Totalmente indipendente dal mondo OO** (e da UML)
- Tecnica **non visuale!**
 - solo testo, formattato in modo standard
 - invece è visuale lo **Use Case diagram (UML)**

USE CASE

- Esprimono i **requisiti funzionali** di un sistema dal **punto di vista dell'utente**
- Il sistema è visto come una **black-box**



ESEMPIO ATM

- 1) L'utente seleziona l'importo da prelevare e conferma la cifra selezionata
- 2) Il sistema controlla che l'importo sia disponibile ed eroga le banconote

- Gli use case esprimono l'**interazione** tra le entità che interagiscono con il sistema stesso (chiamate attori)
 - Interazione = scambio di messaggi

*La differenza sostanziale è nel modo
in cui è presentata l'informazione*

REQUISITI E USE CASE

○ Requisito (funzionale):

descrive una funzionalità, o aspetto di dettaglio di una funzionalità, richiesta dal committente o da altri stakeholder del sistema

- **Punto di vista del sistema**

- Es. Il sistema garantisce l'accesso al sistema se username e password fornite dall'utente sono corrette

○ Caso d'uso:

“storia” che descrive una modalità di utilizzo del sistema da parte di un utilizzatore

- **Punto di vista dell'utente**

ID	UC01
Name	Log in
Brief Description	The user logs in the system.
Actor	Consumer user

Main Flow

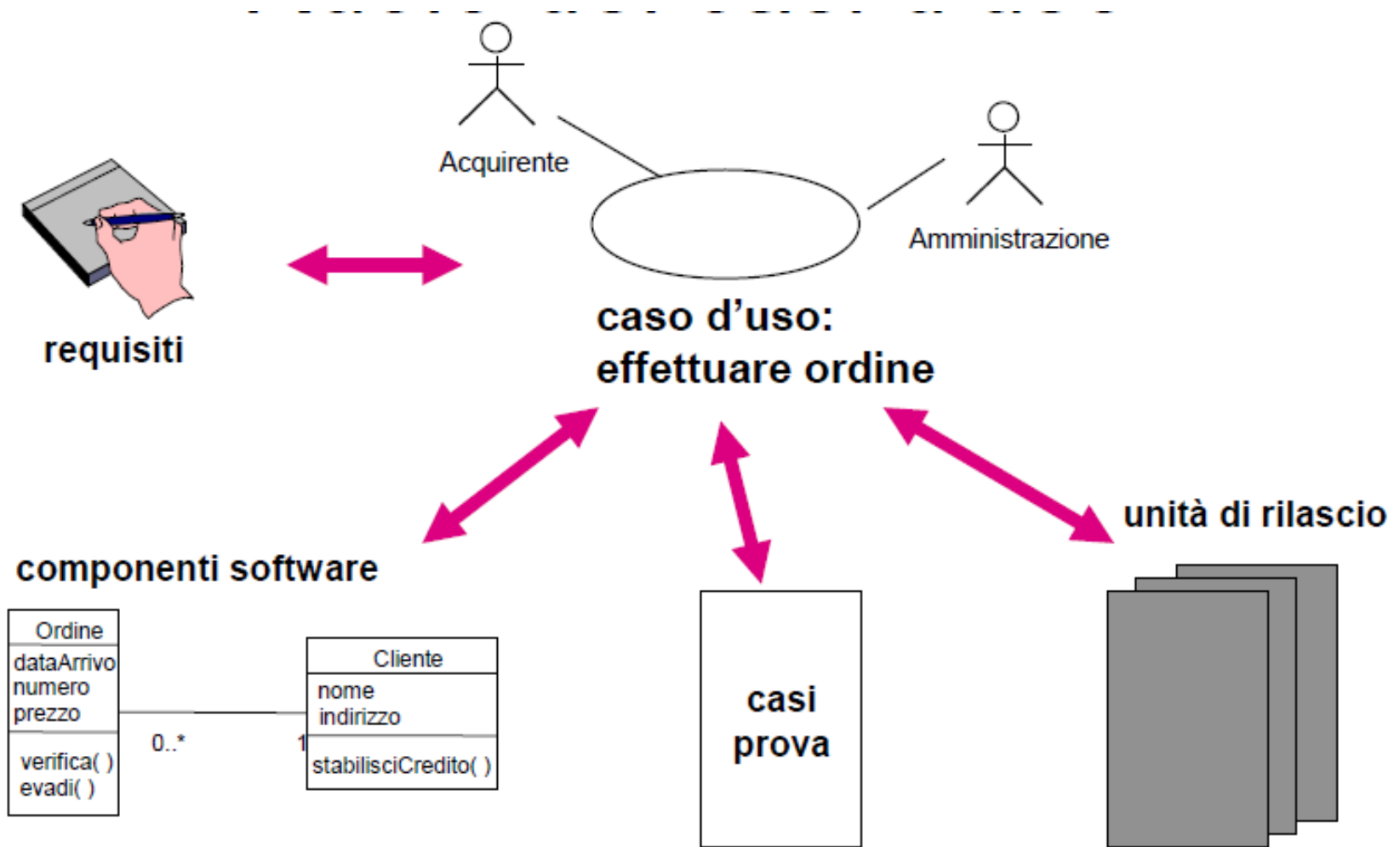
1. This use case starts when the user accesses a protected page. The system displays a login form.
2. The user enters correct ID and password. The system authenticates the user, and displays the protected page requested. The use case ends.

Alternative Flow 1 Starts after Main Flow - step 1

1. The user enters invalid ID or password. The system displays an error message. The use case ends.

Figure 2 "Log In" use case description

UTILI PER?



Modello delle classi (design)

Testing

Planning

DEFINIZIONE DEI REQUISITI BASATA SU USE CASE

○ Composta da:

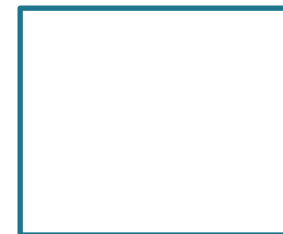
- **Attori** - I ruoli assunti dalle entità (persone, sistemi hardware/software, dispositivi) che interagiscono con il sistema
- **Use case** - quello che gli attori possono fare con il sistema
- **Relazioni** - tra gli attori e gli use case (indica partecipazione!)
- **Confini del sistema** - un rettangolo disegnato intorno agli use case per indicare i confini del sistema

Rappresentazione grafica



Utente

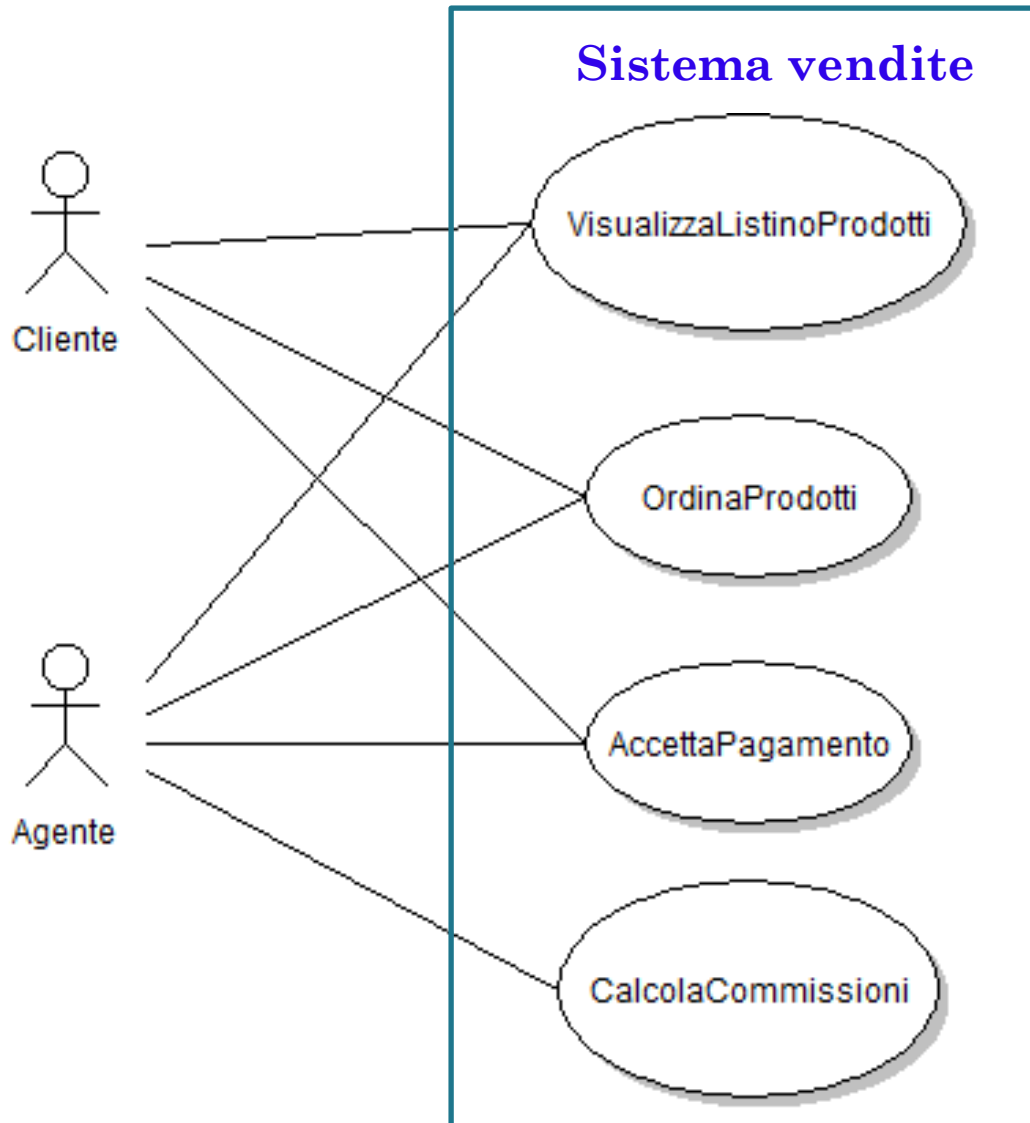
CreaNuovoAccount



ESEMPIO DI DEFINIZIONE (USE CASE DIAGRAM + USE CASES)

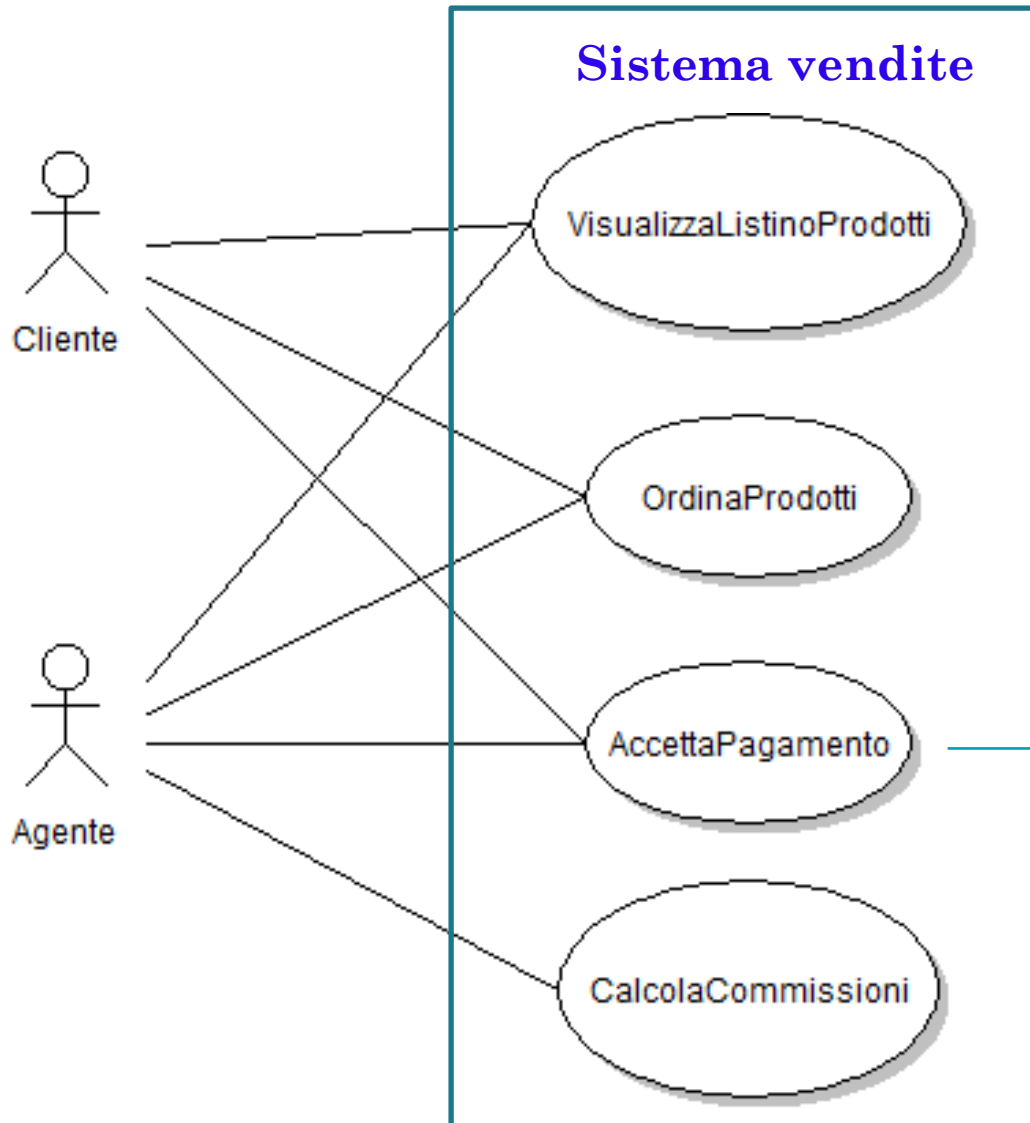
Use Case Diagram (UML)

Sistema composto da quattro use case



ESEMPIO DI DEFINIZIONE (USE CASE DIAGRAM + USE CASES)

Use Case Diagram (UML)

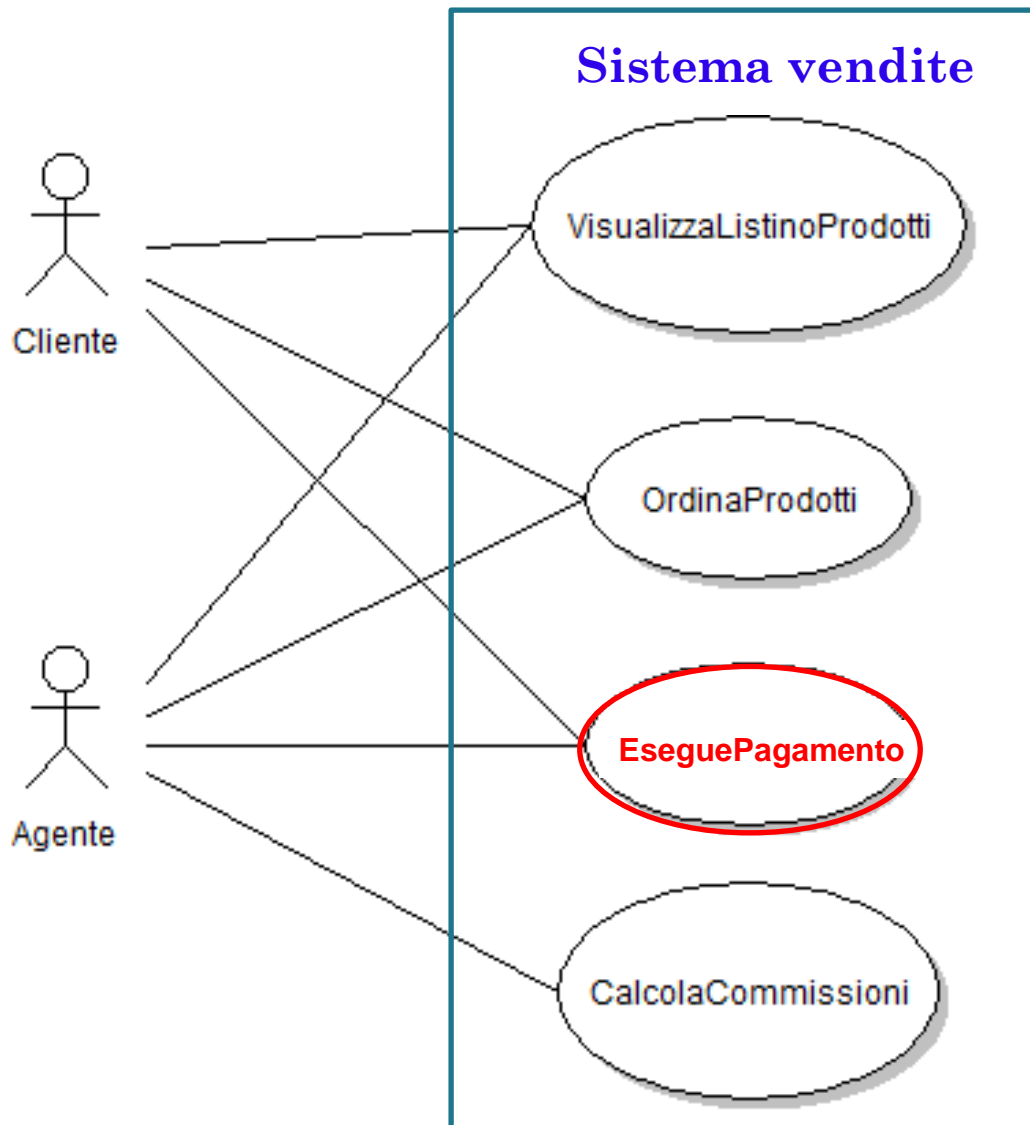


Imprecisione!

EseguePagamento

ESEMPIO DI DEFINIZIONE (USE CASE DIAGRAM + USE CASES)

Use Case Diagram (UML)



Descrizioni degli Use Cases

VisualizzaListinoProdotti

....

....

....

OrdinaProdotti

....

....

....

EseguePagamento

....

....

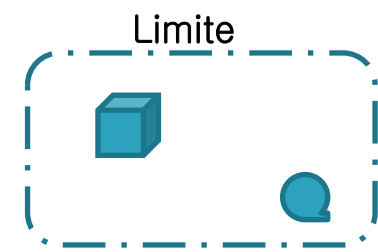
....

CalcolaCommissioni

....

....

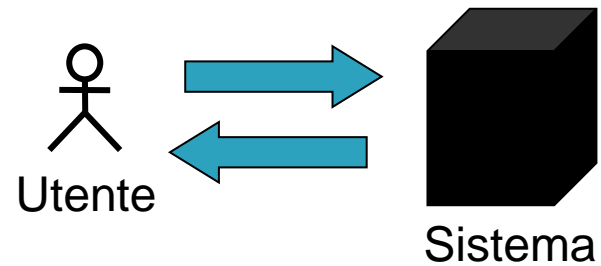
....



LIMITI DEL SISTEMA

- Quando si vuole costruire un sistema la prima cosa da fare è **stabilire** i suoi **limiti/confini** ...
 - Cosa fa parte del sistema (**dentro i confini**) e cosa no (**fuori dai confini**)
- Il posizionamento del confine ha un grosso impatto sui requisiti funzionali
- Molti problemi a progetti sono sorti proprio a causa di un confine non ben definito ...
 - Bisogna chiarire ciò che già esiste all'esterno del sistema (ad esempio altri sistemi) e cosa deve essere svolto dal sistema che vogliamo sviluppare!!!

ATTORI



- Gli attori rappresentano **chi/cosa interagisce** con il sistema
- Un attore rappresenta **un ruolo** che un'entità esterna “recita” interagendo con il sistema
 - Es. un utente, un congegno hardware (es. sensore) o un altro sistema
- Un ruolo è come un cappello che una persona indossa in un contesto particolare
 - Non confondere un ruolo con la cosa stessa!!!
 - Es. **Attore utente**: molte persone possono giocare quel ruolo (filippo, gianni, gigi) ma possono anche ricoprirne altri. Ad esempio filippo potrebbe essere anche **amministratore del sistema**

CATEGORIE DI ATTORI

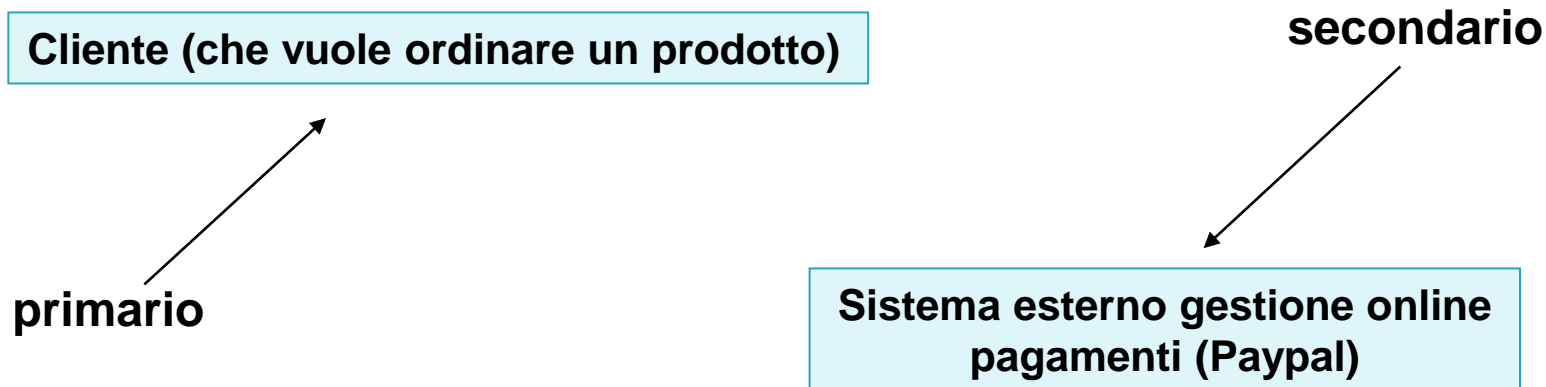
- **Primari**

- chi ha delle mire sul sistema
- chi guadagna qualcosa dal sistema

- **Secondari**

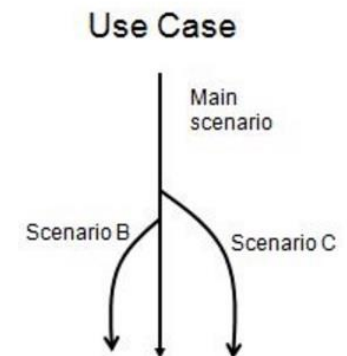
- quelli su cui il sistema ha delle mire
- chi produce qualcosa (o offre un servizio) per il sistema

Applicazione Web Amazon



SCENARIO

- Uno scenario è una **sequenza ordinata di interazioni** tra un sistema e gli attori
 - **Sequenza ordinata di msg**
- Rappresenta una particolare esecuzione di uno use case (**istanza**), e rappresenta un singolo cammino dello use case
- Gli scenari sono utili poiché:
 - È più facile partire con scenari (concreti) e poi generalizzare (e quindi definire lo use case)
 - Sono usati per il testing



SCENARIO

Esempio ATM



Uno scenario:

- 1) L'utente si autentica con carta e PIN
- 2) Il sistema ne verifica la correttezza e conferma
- 3) L'utente seleziona l'importo da prelevare e conferma la cifra selezionata
- 4) Il sistema controlla che l'importo sia disponibile ed eroga le banconote

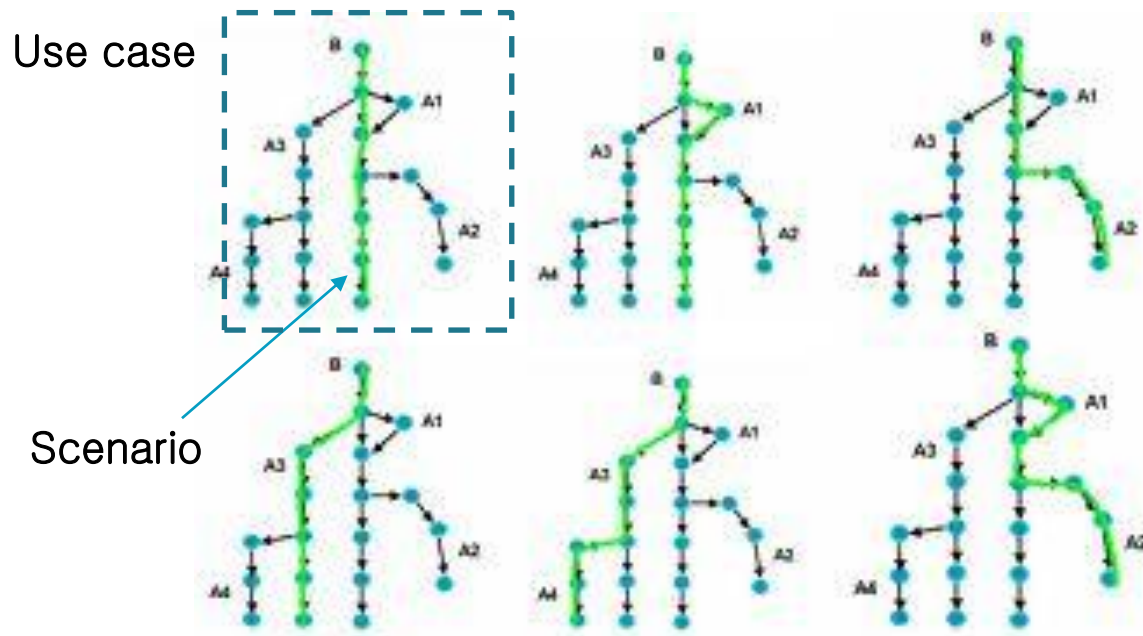
○ Ma non è l'unico modo in cui potrebbero andare le cose:

- Numero di conto o PIN potrebbero non essere corretti
- La cifra richiesta potrebbe non essere disponibile sul conto, o non essere disponibile nello sportello
- L'utente potrebbe non confermare **o voler richiedere la ricevuta**

○ Si hanno **diversi scenari**, ma in tutti l'utente ha lo stesso scopo: **prelevare del contante** (che è lo use case ...)

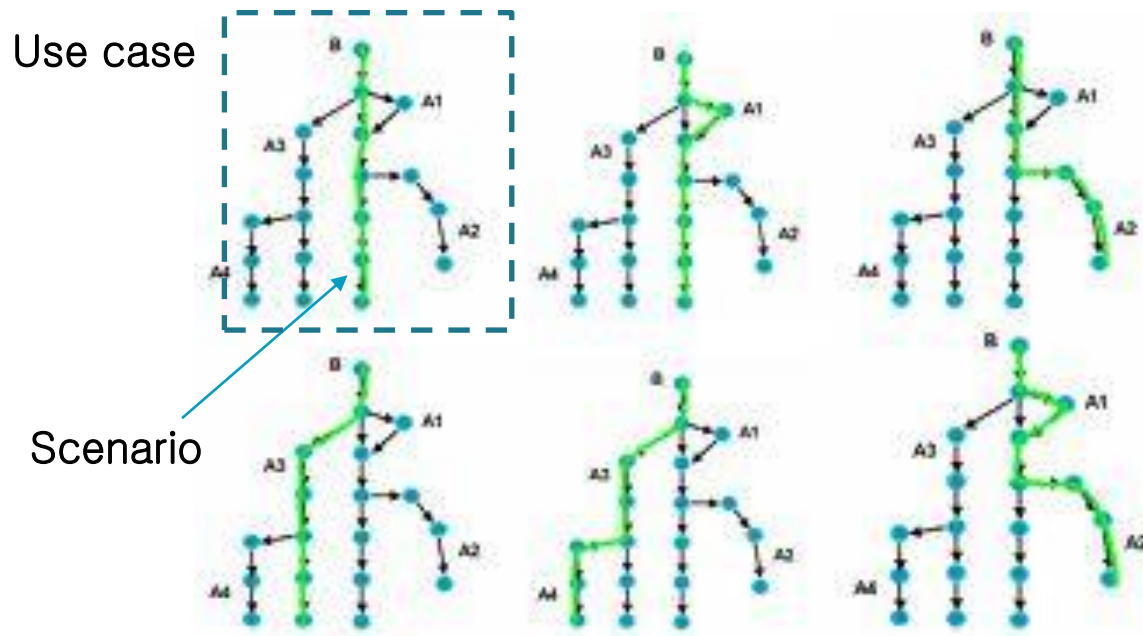
USE CASE

- È un **insieme di scenari** che hanno in comune lo scopo finale dell'utente
 - Esempio: prelevare contante
- **Relazione tra use case e scenario:**
 - Uno scenario rappresenta un singolo cammino attraverso lo use case



USE CASE

- È un **insieme di scenari** che hanno in comune lo scopo finale dell'utente
 - Esempio: prelevare contante
- **Relazione tra use case e scenario:**
 - Uno scenario rappresenta un singolo cammino attraverso lo use case



DESCRIVERE UNO USE CASE (1)

- Gli use case in genere sono dati come **testo strutturato**
 - **Form/template**
- I passi di uno use case sono **testo facile da capire**
- Viene usato il vocabolario del **dominio dell'applicazione**
 - Il cliente deve essere in grado di comprenderli ...
- Gli use case sono descrizioni chiare, precise, generali, e indipendenti dalle tecnologie
 - Ricordarsi le **proprietà dei requisiti** ...

Use Case 1.1 *Withdraw money from an ATM*

Withdraw money from an ATM

Primary actors: Customer

Preconditions: Network connection is active
ATM has available cash

Basic flow of events:

1. Bank customer inserts debit card and enters PIN.
2. Customer is validated.
3. ATM displays actions available on ATM unit. Customer selects Withdraw Cash.
4. ATM prompts account.
5. Customer selects account.
6. ATM prompts amount.
7. Customer enters desired amount.
8. Information sent to Bank, inquiring if sufficient funds/allowable withdrawal limit.
9. Money is dispensed and receipt prints.

Alternative flows

- 2a. Customer is not validated.
 - 2a1. ATM displays error message.
- 7a. Customer selects invalid amount.
 - 7a1. ATM prompts user to re-enter valid amount.
- 8a. Customer has insufficient funds.
 - 8a1. ATM displays error message.
 - 8a2. ATM shows available withdrawal limit, redirects to step 6.
- 9a. ATM has insufficient cash.
 - 9a1. ATM Technician is alerted.
 - 9a2. ATM displays error message and phone number to call.
- 9b. Cash gets stuck in dispensing.
 - 9b1. ATM displays error message.

Esempio di Use case

DESCRIVERE UNO USE CASE (2)

- Scenario principale: normale flusso di eventi
 - Scenario del **mondo perfetto** ...
 - in cui tutto avviene come previsto/desiderato
- Scenari secondari: cosa può succedere di sbagliato/diverso e come gestirlo
 - Condizioni di errore
 - Casi particolari
 - Altre possibili azioni/behavior

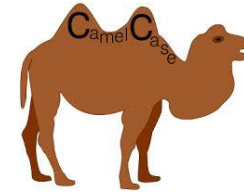


Use Case Template

- pattern che le descrizioni degli use case devono seguire
- molte varianti, senza differenze essenziali
- quella che utilizzeremo è usata nel libro

UML 2 e Unified Process, Jim Arlow e Ila Neustadt

USE CASE TEMPLATE



InserisceProdottiLocali

Nome dello use case: è il goal dello use case

“breve frase verbale attiva” --> UpperCamelCase

Identificatore: di solito numerico progressivo

Breve descrizione: un paragrafo che fissa l'obiettivo dello use case

Attori primari: l'attore/gli attori primari dello use case

Attori secondari: gli attori che “servono” per svolgere lo use case

Precondizioni: vincoli sullo stato corrente del sistema

Scenario principale: i passi che costituiscono lo use case

Postcondizioni: condizioni che devono essere vere quando lo use case termina con successo l'esecuzione

Scenari alternativi: un elenco di alternative allo scenario principale

PASSI DI UNO SCENARIO

- Uno scenario è costituito da un **elenco di passi** che devono essere concisi, numerati e ordinati temporalmente
- **Passo:**
 - <numero> Il <qualcosa/qualcuno> <qualche azione>

Use Case 1.1 *Withdraw money from an ATM*

Withdraw money from an ATM

Primary actors: Customer

Preconditions: Network connection is active
ATM has available cash

Basic flow of events:

1. Bank customer inserts debit card and enters PIN.
2. Customer is validated.
3. ATM displays actions available on ATM unit. Customer selects Withdraw Cash.
4. ATM prompts account.
5. Customer selects account.
6. ATM prompts amount.
7. Customer enters desired amount.
8. Information sent to Bank, inquiring if sufficient funds/allowable withdrawal limit.
9. Money is dispensed and receipt prints.



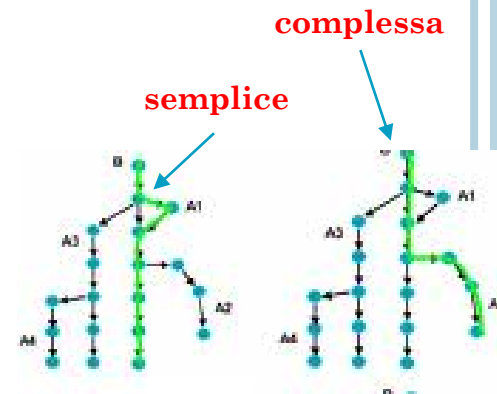
DEVIAZIONI E RIPETIZIONI



- Si ha una deviazione tutte le volte che ci si allontana dallo **scenario principale**

- Deviazioni:**

- Semplici:** usare “se” nella sequenza principale
 - vedi esempio pagina successiva
- Complesse:** scrivere sequenze degli eventi alternative



Le deviazioni complesse di solito rappresentano errori o casi particolari che non ritornano sullo scenario principale

- Ripetizioni all'interno di una sequenza**

- Parole chiave
 - Per e Fintantochè



ESEMPIO: USO DEL SE-ALTRIMENTI

*Template «Jim Arlow»
adottato nel corso*

Caso d'uso: TrovaLibro_se

Id: 6

Breve descrizione: il cliente ricerca un libro nella **biblioteca on-line**

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: nessuna (o autenticazione)

Sequenza degli eventi principali

1. Il cliente seleziona la funzionalità
2. Il sistema chiede al cliente i criteri di ricerca
3. Il cliente inserisce i criteri di ricerca
4. Il sistema ricerca i libri che corrispondono ai criteri inseriti dal cliente

5. **Se** il sistema trova uno o più libri
5.1 Il sistema mostra l'elenco dei libri

6. **Altrimenti**

6.1 Il sistema comunica che non sono stati trovati libri che soddisfano i requisiti richiesti

Postcondizioni: nessuna

Sequenza degli eventi alternativa: nessuna

ESEMPIO: USO DEL PER

Caso d'uso: TrovaLibro_Per

Id: 6

Breve descrizione: il cliente ricerca un libro nella **biblioteca on-line**

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: nessuna

Sequenza degli eventi principale:

1. Il cliente seleziona la funzionalità “trova libro”
2. Il sistema chiede al cliente i criteri di ricerca
3. Il cliente inserisce i criteri di ricerca e conferma
4. Il sistema ricerca i libri che corrispondono ai criteri di ricerca immessi dal cliente
5. Se il sistema trova uno o più libri
 - 5.1 **Per** ogni libro trovato
 - 5.1.1 Il sistema mostra l'immagine del libro
 - 5.1.2 Il sistema mostra le caratteristiche del libro
 - 5.1.3 Il sistema mostra il prezzo
6. Altrimenti
 - 6.1 Il sistema comunica che non sono stati trovati libri che soddisfano i requisiti richiesti

ESEMPIO: USO DEL FINTANTOCHÉ

Caso d'uso: TrovaLibro_Fintantochè

Id: 6

Breve descrizione: il cliente ricerca un libro nella **biblioteca on-line**

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: nessuna

Sequenza degli eventi principale:

1. Il cliente seleziona “trova libro”
2. Il sistema chiede al cliente i criteri di ricerca
3. Il cliente inserisce i criteri di ricerca
4. **Fintantochè** i criteri di ricerca non sono validi
 - 4.1 Il sistema chiede di ri-inserire i criteri
 - 4.2 Il cliente inserisce i criteri di ricerca
 - 4.3 Il sistema valida i criteri inseriti
5. Il sistema ricerca i libri che corrispondono ai criteri di ricerca immessi dal cliente
6. Se il sistema trova uno o più libri
 - 6.1 Il sistema mostra l'elenco dei libri
7. Altrimenti
 - 7.1 Il sistema comunica che non sono stati trovati libri

SCENARI ALTERNATIVI

- Di solito sono documentati a parte per semplificare gli use case

Caso d'uso: CreaNuovoAccount

Id: 9

Breve descrizione: il sistema **crea un nuovo account** per il cliente

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: nessuna

Sequenza degli eventi principale:

1. Il cliente seleziona “**crea nuovo account**”
2. Il sistema chiede di inserire le info del cliente, tra cui e-mail, parola chiave e parola chiave di conferma
3. Il cliente inserisce le info e conferma
4. Il sistema valida le info inserite
5. Il sistema crea un nuovo account per il cliente

Postcondizioni: un nuovo account è stato creato per il cliente

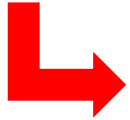
Sequenza degli eventi alternativa:

IndirizzoPostaElettronicoNonValido, ParolaChiaveNonValida, Annulla



SPECIFICARE GLI SCENARI ALTERNATIVI

Relativo allo use case 9



**Sequenza degli eventi alternativa: CreareNuovoAccount:
IndirizzoPostaElettronicaNonValido**

Id: 9.1

Breve descrizione: il sistema informa il cliente che ha inserito un indirizzo e-mail non valido

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: Il cliente ha inserito un indirizzo non valido

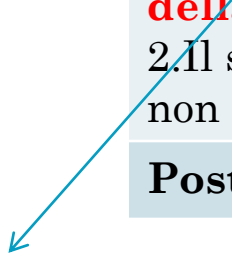
Sequenza degli eventi principale:

1. La sequenza degli eventi alternativa inizia dopo il passo 4 della sequenza degli eventi principale

2. Il sistema informa il cliente che ha inserito un indirizzo di posta non valido

Postcondizioni: nessuna

Specifica la “relazione”
con lo use case
principale



E' anche possibile che la sequenza di eventi possa essere attivata in qualunque momento della sequenza principale

(Vedere slide successiva)

ATTIVAZIONE IN QUALUNQUE MOMENTO: ESEMPIO

Sequenza degli eventi alternativa: CreaNuovoAccount:
ANNULLA

Id: 9.2

Breve descrizione: il cliente annulla il processo di creazione dell'account

Attori primari: Cliente

Attori secondari: nessuno

Precondizioni: nessuna

Sequenza degli eventi principale:

1.La sequenza degli eventi alternativa inizia in qualsunque momento in cui l'utente deve compiere l'azione

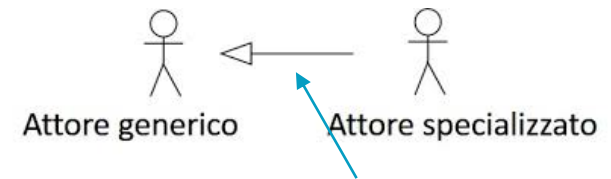
2.Il cliente annulla la creazione dell'account

3.Il sistema mostra la pagina principale

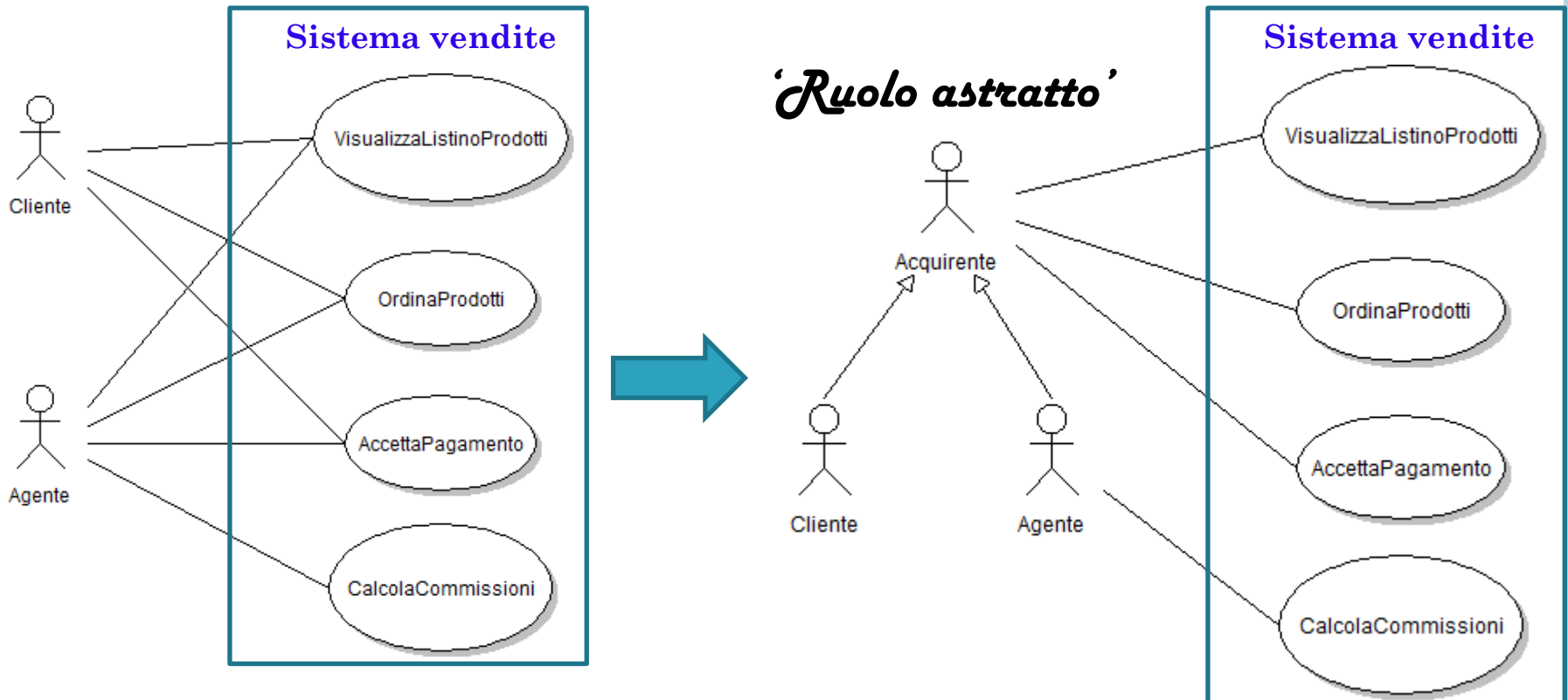
Postcondizioni: nessuna

GERARCHIE DI ATTORI

È possibile definire **gerarchie di attori**



*Generalizzazione/
Specializzazione*

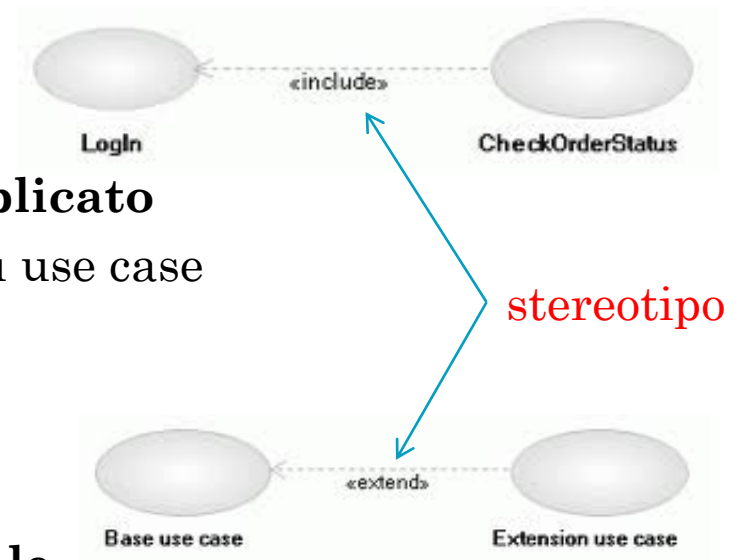


Cliente e Agente ereditano tutte le relazioni di Acquirente!

RELAZIONI TRA USE CASE

- **Inclusione (<<include>>)**

- Per decomporre uno use case complicato
- **Comportamento comune** a uno o più use case

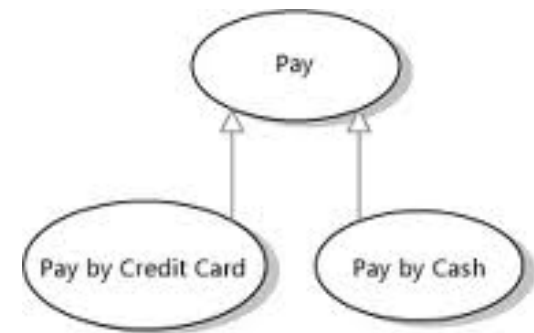


- **Estensione (<<extend>>)**

- Mostra un comportamento opzionale

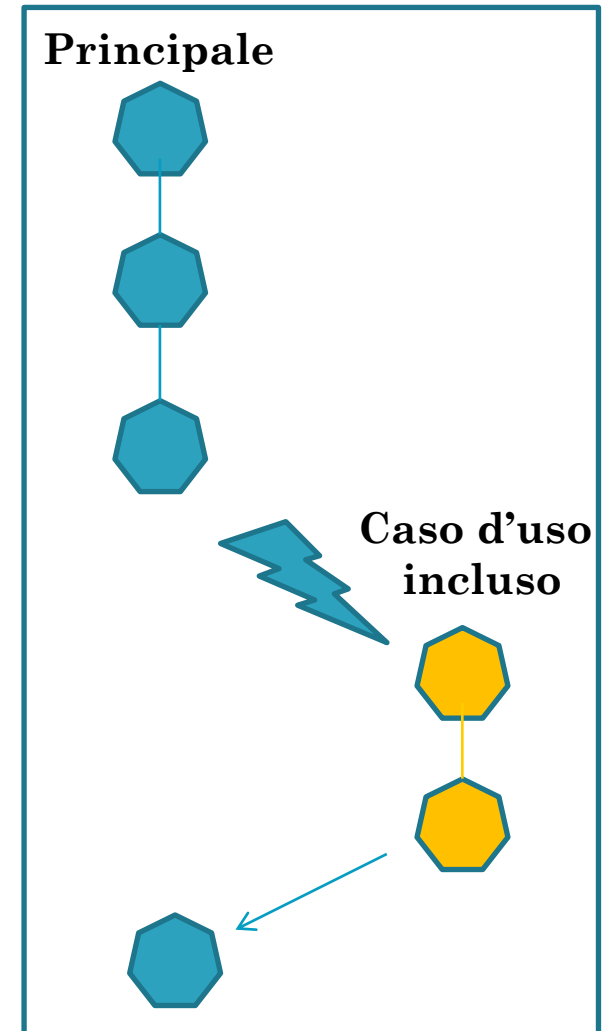
- **Generalizzazione/specializzazione**

- Per specializzare uno use case
 - Es. 'Pay by Cash' specializza 'Pay'

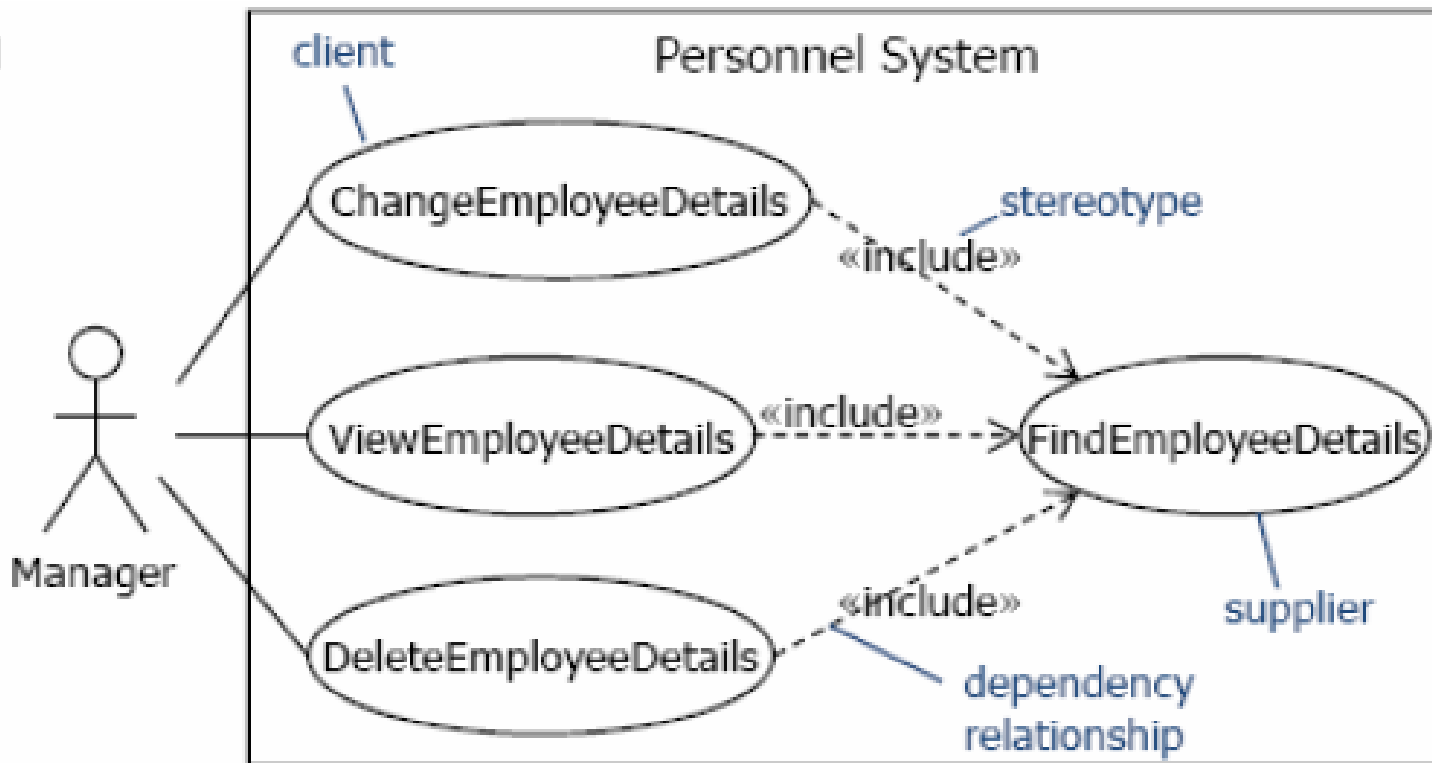


INCLUDE

- Assomiglia al concetto di **procedura/funzione/macro ...**
- Lo use case “**principale**” esegue i passi fino al punto di inclusione e passa il controllo allo use case incluso
 - Alla fine il controllo ritorna allo use case principale
- Lo use case principale senza use case incluso risulta incompleto
- Lo use case incluso può essere completo (e cioè istanziabile) oppure essere un frammento



ESEMPIO DI INCLUDE



COME SI RAPPRESENTA?

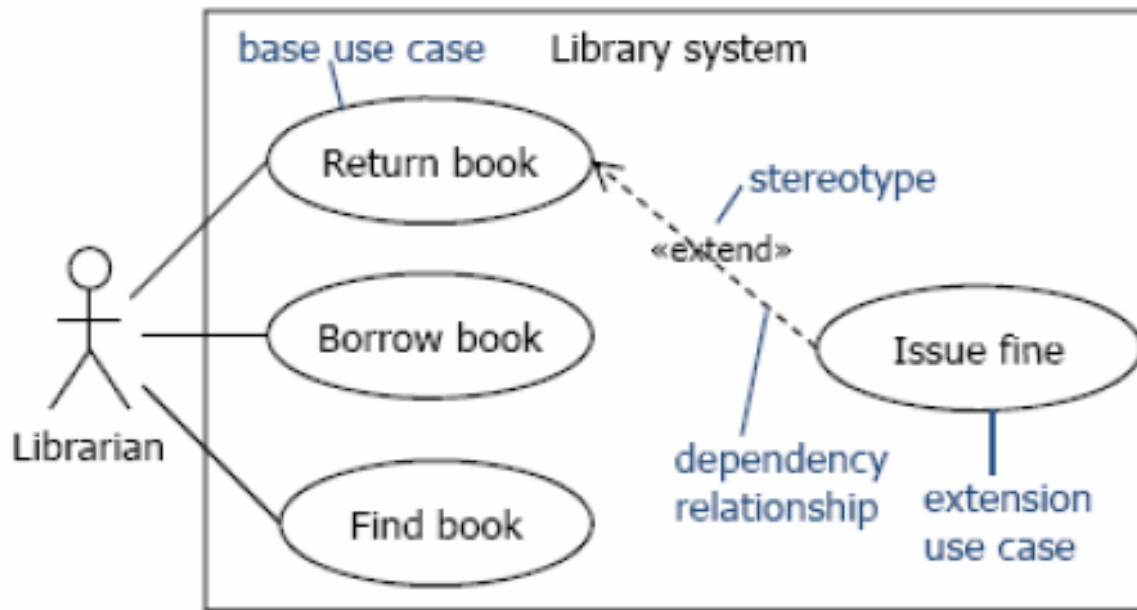
ID:	1	Nome Use Case: ChangeEmployeeDetails
Breve descrizione:	The Manager changes the employee details	
Attori primari:	Manager	
Attori secondari:	Nessuno	
Precondizioni:	1.The Manager is logged on to the system.	
<div>Flusso Principale:</div> <div>1.include(FindEmployeeDetails)</div> <div>2.The system displays the employee details.</div> <div>3.The Manager changes the employee details.</div> <div>4.....</div>		
Postcondizioni:	1. The employee details have been changed	
Flussi Alternativi:	Nessuno	
Punti di estensione:	Nessuno	

ID:	4	Nome Use Case: FindEmployeeDetails
Breve descrizione:	The Manager finds the employee details	
Attori primari:	Manager	
Attori secondari:	Nessuno	
Precondizioni:	1.The Manager is logged on to the system.	
Flusso Principale: 1.The Manager enters the employee's ID. 2.The system finds the employee details.		
Postcondizioni:	1.The system has found the employee details.	
Flussi Alternativi:	Nessuno	
Punti di estensione:	Nessuno	

1.include(FindEmployeeDetails)

Sistema per la gestione di una Biblioteca

EXTEND



*“Issue fine” =
emissione multa*

- Per estendere il comportamento di uno use case ...
- Lo use case base definisce un **punto di estensione** ma non conosce il comportamento dell'estensione
- Lo use case base è completo anche senza estensione
- Lo use case di estensione è solitamente incompleto

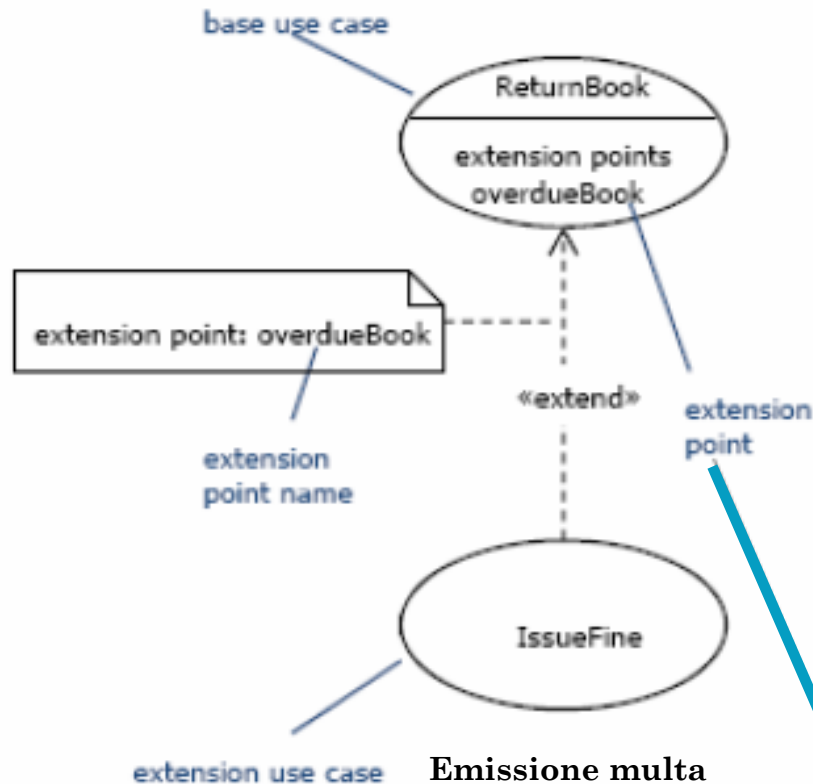
'overdue' = scaduto

ESEMPIO: USE CASE BASE

ID:	9	Nome Use Case: ReturnBook
Breve descrizione:	The Librarian returns a borrowed book.	
Attori primari:	Librarian (bibliotecario)	
Attori secondari:	Nessuno	
Precondizioni:	1.The Librarian is logged on to the system.	
<div>Flusso Principale:</div> <div>1.The Librarian enters the borrower's ID number.</div> <div>2.The system displays the borrower's details including the list of borrowed books.</div> <div>3.The Librarian finds the book to be returned in the list of books. Extension point: overdueBook</div> <div>4.The Librarian returns the book</div>		
Postcondizioni:	1.The book has been returned	
Flussi Alternativi:	Nessuno	
Punti di estensione:	•after step 3: overdueBook	

'overdue' = scaduto

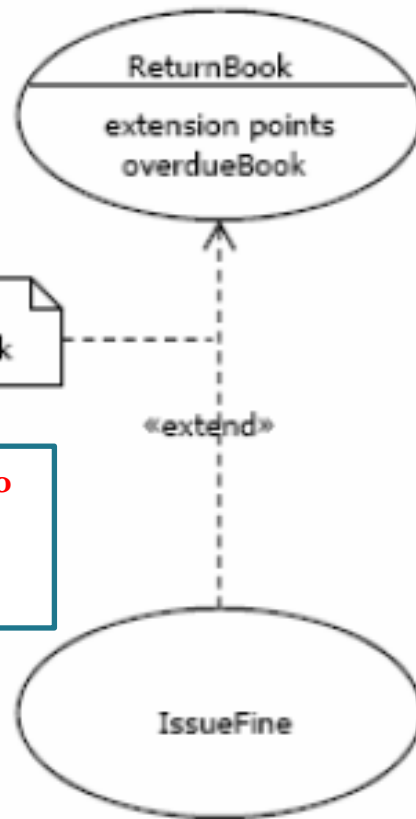
ESEMPIO: USE CASE BASE



ID:	9	Nome Use Case: ReturnBook
Breve descrizione:	The Librarian returns a borrowed book.	
Attori primari:	Librarian (bibliotecario)	
Attori secondari:	Nessuno	
Precondizioni:	1.The Librarian is logged on to the system.	
<p style="text-align: center;">Flusso Principale:</p> <p>1.The Librarian enters the borrower's ID number.</p> <p>2.The system displays the borrower's details including the list of borrowed books.</p> <p>3.The Librarian finds the book to be returned in the list of books. Extension point: overdueBook</p> <p>4.The Librarian returns the book</p>		
Postcondizioni:	1.The book has been returned	
Flussi Alternativi:	Nessuno	
Punti di estensione:	•after step 3: overdueBook	

ESEMPIO: USE CASE DI ESTENSIONE

'overdue' = scaduto



I passi di IssueFine sono inseriti dove c'è l'extension point overdueBook

Emissione multa

ID:	10	Nome useCase di estensione: IssueFine
Breve descrizione:	The Librarian records and prints out a fine	
Attori primari:	Librarian (bibliotecario)	
Attori secondari:	Nessuno	
Precondizioni:	1. The returned book is overdue.	
Flusso Principale: 1. The Librarian enters details of the fine into the system. 2. The system prints out the fine.		
Postcondizioni:	1. The fine has been recorded in the system. 2. The system has printed out the fine.	
Flussi Alternativi:	Nessuno	
Punti di estensione:	Nessuno	

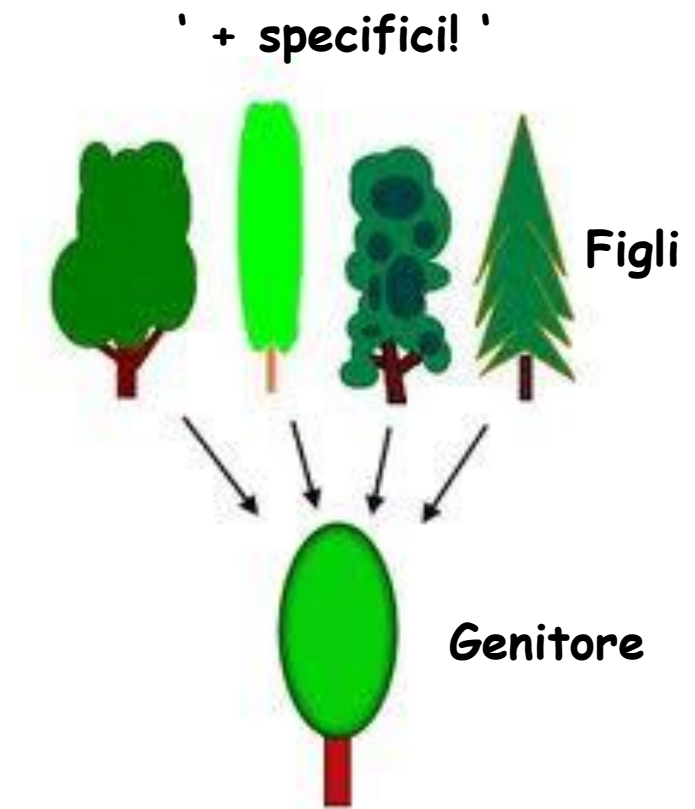
GENERALIZZAZIONE / SPECIALIZZAZIONE

- Gli use case specializzati (figli) rappresentano delle **varianti più specifiche** dello use case generalizzato (genitore) da cui ereditano

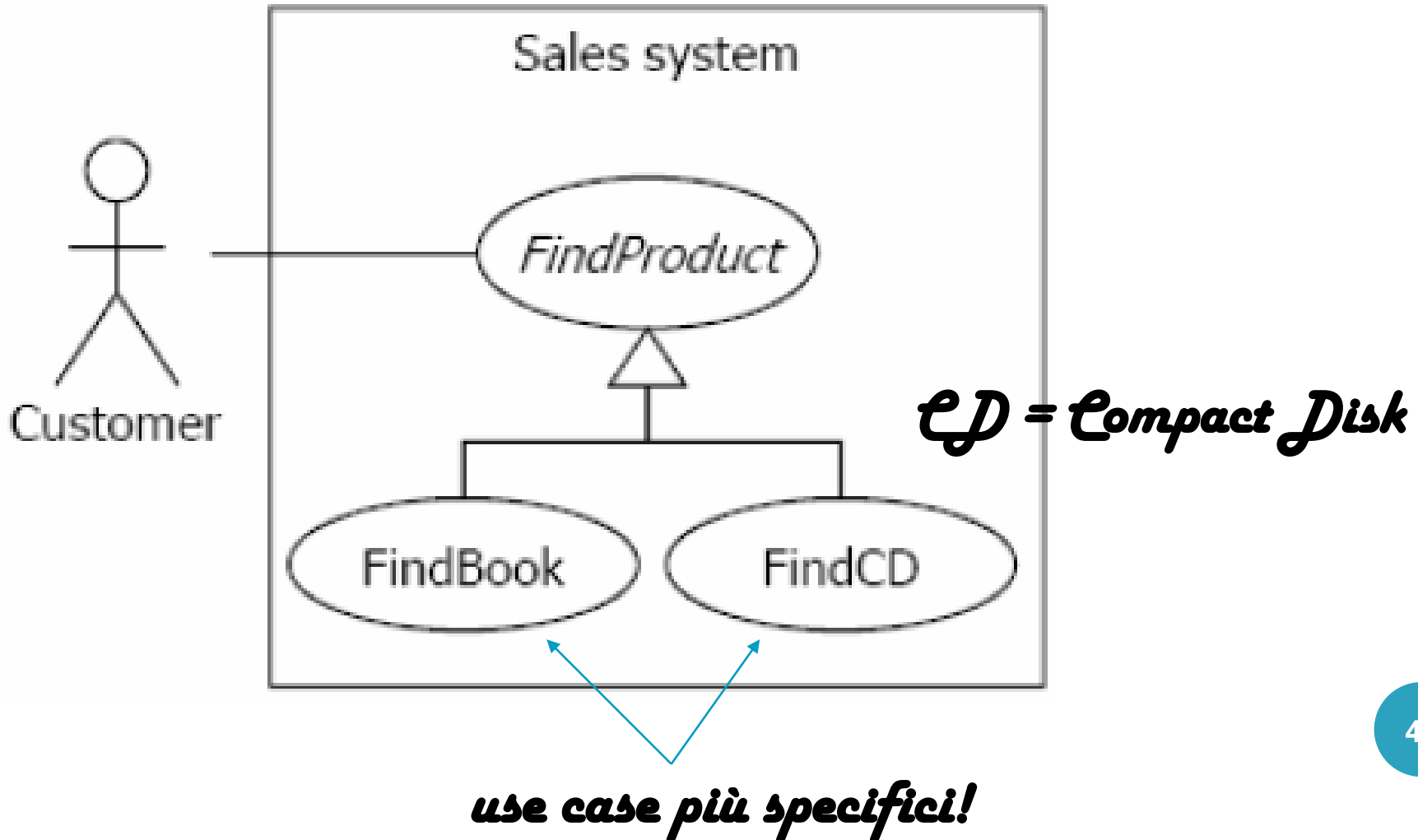
- Es. **TrovaLibro** specializza **TrovaProdotto**

- Gli use case ‘figli’ possono:

- Ereditare i passi del genitore
 - Aggiungere nuovi passi
 - Ridefinire (modificare) i passi ereditati



ESEMPIO DI GENERALIZZAZIONE



ESEMPIO DI GENERALIZZAZIONE

NEWS

ADDIO AL COMPACT DISC, DAL 2022 STOP ALLA PRODUZIONE

18 giugno 2020



isk

COME SI RAPPRESENTANO I PASSI MODIFICATI, AGGIUNTI ED EREDITATI?

Esistono convenzioni!

FindProduct:

1. Il cliente seleziona la funzionalità “trova prodotto”
2. Il sistema chiede al cliente i criteri di ricerca
3. Il cliente inserisce i criteri di ricerca
4. Il sistema ricerca i prodotti che corrispondono ai criteri di ricerca immessi dal cliente
5. **Se** il sistema trova uno o più prodotti
 - 5.1 Il sistema mostra l'elenco dei prodotti
6. **Altrimenti**
 - 6.1 Il sistema comunica che non sono stati trovati prodotti che soddisfano i requisiti richiesti

Eredita “as is”

Eredita e modifica (override)

FindBook:

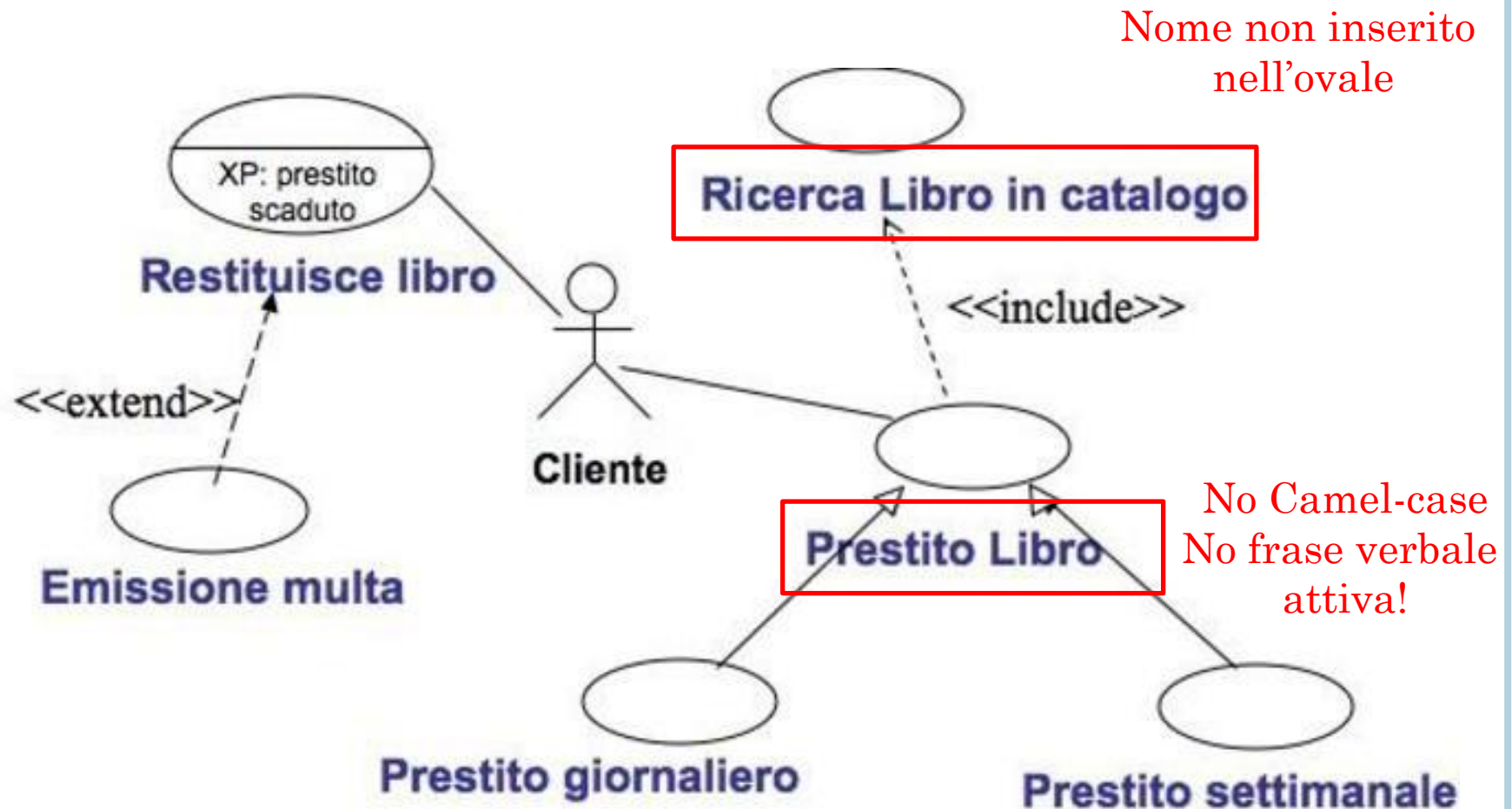
1. (o1.) Il cliente seleziona la funzionalità “trova **libro**”
2. (o2.) Il sistema chiede al cliente i criteri di ricerca, **ovvero autore, titolo e ISBN**
3. (3) Il cliente inserisce i criteri di ricerca
4. (o4.) Il sistema ricerca i **libri** che corrispondono ai criteri di ricerca immessi dal cliente
5. (o5.) **Se** il sistema trova uno o più **libri**
 - (o5.1) Il sistema mostra l'elenco dei **libri** visualizzando **autore e titolo**
6. (6) **Altrimenti**
 - 6.1 il sistema visualizza l'attuale best-seller
 - 6.2 (6.1) Il sistema comunica che non sono stati trovati prodotti che soddisfano i requisiti richiesti

Aggiunta

Eredita “as is”
+ rinumerazione

Problema: poco chiare in particolare per non esperti (es. stakeholders)

ESEMPIO: SISTEMA GESTIONE BIBLIOTECA

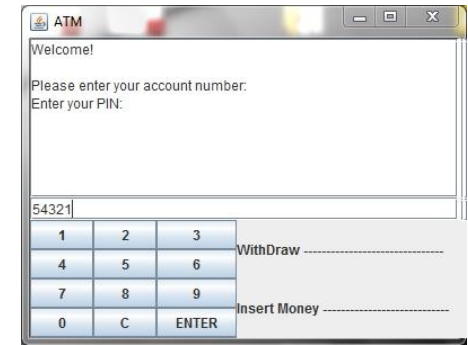


che problemi ha? errore grave ... Confini!

SCREEN MOCKUPS

- Spesso gli use case “sono completati” da **Screen Mockups** (“sketch” dell’interfaccia utente) per chiarirne il comportamento

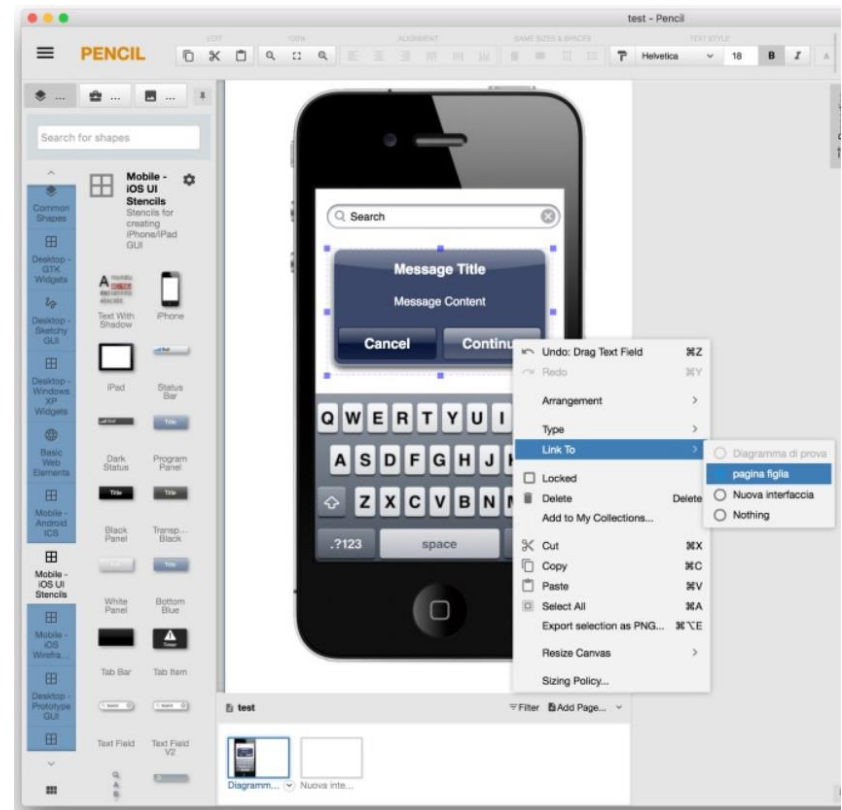
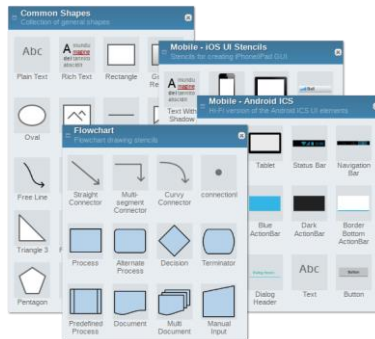
- Utili durante la negoziazione con il cliente
- Utili per lo sviluppatore
- Utili per specificare UI
 - Se dettagliati



- Sono dei disegni che mostrano l’interfaccia utente ad un certo passo dello use case**
- Chiaramente: a fronte di un “migliore comprensione” c’è un effort nel produrli
- Possono essere prodotti su carta o utilizzando un tool specifico (**GUI prototyping tools**)

PENCIL TOOL

- Free and open-source **GUI prototyping tool** usato per creare screen mockups
- Fornisce differenti tipi di **Stencil** per:
 - Desktop, Web e mobile



SCREEN MOCKUPS: ESEMPIO

Use case: Insert Coin Type

Level: User-goal

Primary actor: Coin collector

Intention in context: The Coin collector wants to insert a new type of coin in the Catalogue

Precondition: A non-empty list of issuers of coins is displayed

Main success scenario:

1. The Coin collector selects an issuer of coins from the provided list and asks to insert a new coin type.
2. The System shows all the coin types for the selected issuer and asks the Coin collector to insert information (** 2) for the new coin type (insertCoinTypeMockup).
3. The Coin collector inserts the information.
4. The System asks the Coin collector to confirm the information added
5. The Coin collector confirms
6. The System informs the Coin collector that the insertion was successful (showCoinInformationMockup). The use case ends successfully

Extensions:

- 5.a The Collector does not accomplish the operation. The use case fails.

The mockup shows a window titled "EasyCoin - Insert Coin Type". It has a section "Issuer of coins:" with fields for "Name:" (Lithuania), "Original Name:" (Lietuva), "From:" (2010), "To:" (2011), and "Mint:" (Lithuania state mint). Below this is a list of coin types. The first coin type is selected, showing fields for "Unit:" (Lita), "Value:" (50), "Material:" (Silver Ag 925), "Weight:" (28.28 g), and "Diameter:" (38.61 mm). The second coin type is also visible, showing "Unit:" (Lita), "Value:" (10), "Material:" (Aconital), "Weight:" (27 g), and "Diameter:" (37 mm). There is an "Insert" button at the bottom right.

The mockup shows the same "EasyCoin - Insert Coin Type" window, but with a "Confirmation" dialog box overlaid. The dialog box has a blue header and a message: "1 The coin type has been inserted". There is an "OK" button at the bottom of the dialog box. The background window is partially obscured by the dialog box.

Issuer = ente emettitore

ESPERIMENTO CONTROLLATO



Assessing the Effect of Screen Mockups on the Comprehension of Functional Requirements

FILIPPO RICCA, DIBRIS, University of Genova
GIUSEPPE SCANNIELLO, DiMIE, University of Basilicata
MARCO TORCHIANO, Politecnico di Torino
GIANNA REGGIO, DIBRIS, University of Genova
EGIDIO ASTESIANO, DIBRIS, University of Genova

Over the last few years, the software engineering community has proposed a number of modeling methods to represent functional requirements. Among them, use cases are recognized as an easy to use and intuitive way to capture and define functional requirements. To improve the comprehension of functional requirements, screen mockups (also called user-interface sketches or user interface-mockups) can be used in conjunction with use cases. In this paper, we aim at quantifying the benefits achievable by augmenting use cases with screen mockups in the comprehension of functional requirements with respect to effectiveness, effort, and efficiency. For this purpose, we conducted a family of four controlled experiments, involving 139 participants having different profiles. The experiments involved comprehension tasks performed on the requirements documents of two desktop applications. Independently from the participants' profile, we found a statistically significant large effect of the presence of screen mockups on both comprehension effectiveness and comprehension task efficiency. While no significant effect was observed on the effort to complete tasks. The main "take away" lesson is that screen mockups are able to almost double the efficiency of comprehension tasks.

Categories and Subject Descriptors: D.2 [Software]: Software Engineering; D.2.1 [Software Engineering]: Requirements/Specifications

General Terms: Documentation, Experimentation, Human Factors

Additional Key Words and Phrases: Screen Mockups, Use Cases, Analysis models, Controlled experiment, Family of experiments, Replicated experiments

ACM Reference Format:

ACM Trans. Softw. Eng. Methodol. V, N, Article A (January YYYY), 37 pages.
DOI = 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

1. INTRODUCTION

It is widely recognized that a substantial portion of software defects (up to 85%) originates in the requirements engineering phase of the software development process [Young 2001]. Defects originated in this phase are typically caused by ambiguous, incomplete, inconsistent, silent (unexpressed), unusable, over-specific and verbose requirements (both functional and non-functional). Defects might also stem from communication problems among stakeholders [Meyer 1985].

To face these issues, a number of methods/techniques have been proposed for representing functional and non-functional requirements. With regard to functional requirements, use cases are a well known and widely used method to specify the purpose of a software system and to produce its description in terms of interactions between actors

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© YYYY ACM 1049-331X/YYYY/01-ARTA \$15.00
DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

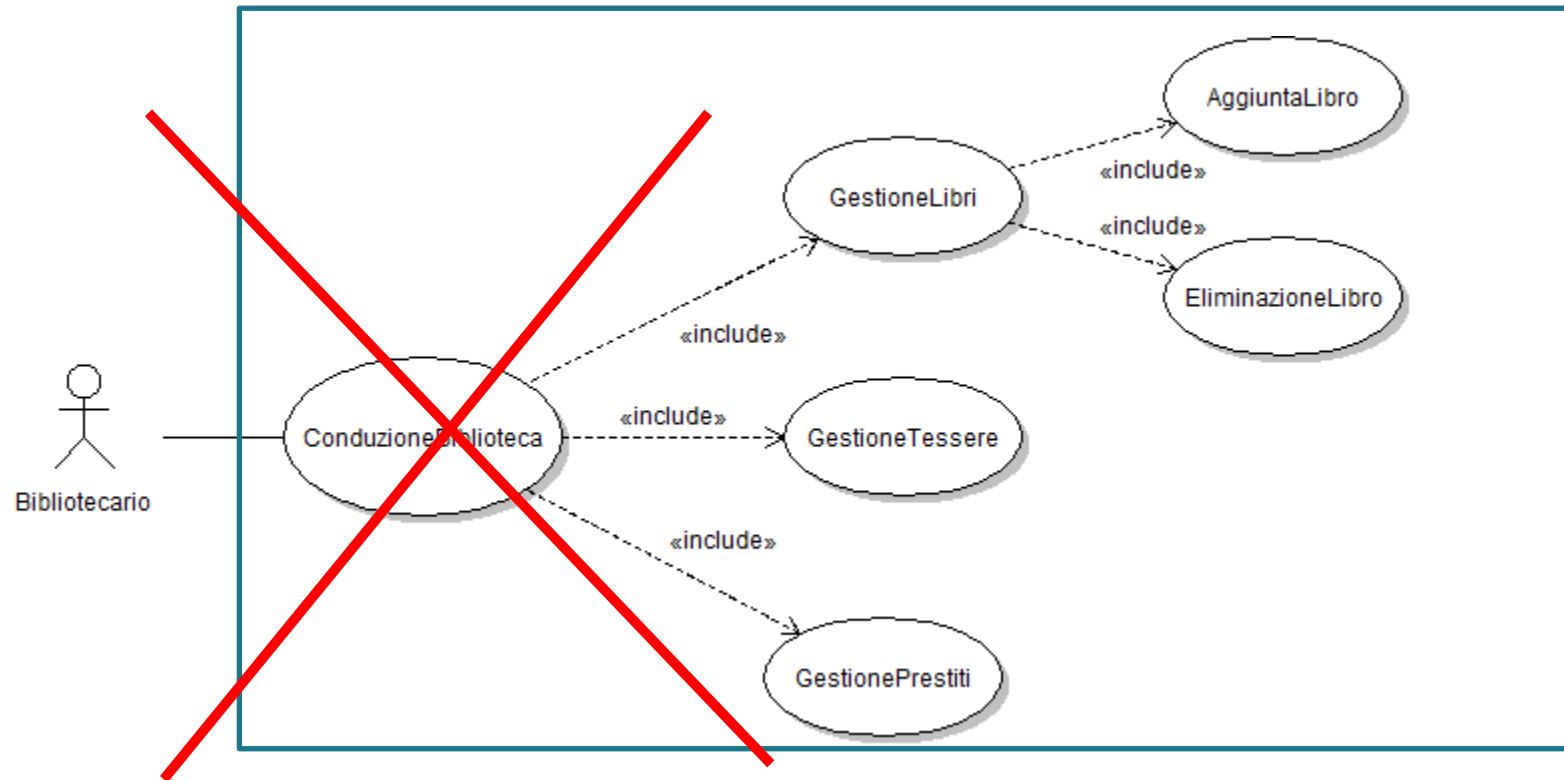
- Usare gli screen mockup:
 - migliora la comprensione degli **Use case** del **+69%**
 - migliora l'efficienza di eseguire task di comprensione del **+89%**

CONSIGLI SU COME SCRIVERE GLI USE CASE

- **Mantenere gli use case brevi e semplici**
 - Scenario principale non deve superare una pagina foglio A4!
- **Non troppo astratti non troppo dettagliati**
- **Non usarli per descrivere le funzionalità interne del sistema** (descrivono interazione!)
- **Usare ‘extend’ e ‘generalizzazione’ solo quando questo semplifica il modello**
 - Ricordarsi che gli use case sono letti da umani!!!
- **Non confondere il “cosa” con il “come”**
 - Es. di use case:
 - 4. Il sistema chiede al cliente di confermare l'ordine
 - 5. Il cliente preme il pulsante OK
- **Evitare la scomposizione funzionale**
 - **Errore:** creare un insieme di use case di alto livello e poi dividerlo in un insieme di use case di livello inferiore

Sbagliato è una scelta di design!

ESEMPIO: GESTIONE BIBLIOTECA



- I livelli + alti sono molto astratti e non aggiungono molto (salvo che non si voglia esplicitare un ordine di esecuzione degli use case). Solo i livelli + bassi sono significativi ...

CONCLUSIONI

- Gli use case sono un **potente mezzo** per esprimere i **requisiti funzionali** di un sistema
 - Starting point per il design
 - Aiuto per pianificare i rilasci
 - Guida per software testing
- **Molto usati in pratica!**
 - Sistemi informativi/gestionali
 - Utilizzati in UP
- Diversi template disponibili
 - Più o meno formali ma pressoché equivalenti
- Sono poco adatti in sistemi dominati da requisiti non funzionali, sistemi ricchi di algoritmi complessi e poveri di interazione utente (no GUI, no use case ☹) ...
 - Es. Sistemi embedded



ULTERIORI APPROFONDIMENTI

- ◉ **Writing Effective Use Cases** by Alistair Cockburn, Addison-Wesley
- ◉ UML 2 e Unified Process - Analisi e progettazione Object-Oriented 2/ed by **Jim Arlow, Ila Neustadt**
- ◉ **Use Cases: the Pros and Cons** by Donald G. Firesmith, <http://www.ksc.com/article7>.
- ◉ Introduzione ai casi d'uso in italiano, <http://www.progettoatena.it/stato/UML03.pdf>