



Università
Ca' Foscari
Venezia

Il documento di specifica



Documento di specifica dei requisiti

- E' il documento ufficiale che contiene ciò che è richiesto agli sviluppatori del sistema
- Include sia **definizione** e **specificazione** dei requisiti
- Requisiti del documento:
 - Deve specificare il **comportamento esterno** del sistema
 - Deve specificare i **vincoli** implementativi
 - Deve essere facilmente **modificabile**
 - Deve servire come riferimento per la **manutenzione**
 - Deve contenere **previsioni sul ciclo di vita** del sistema
 - Deve caratterizzare le risposte accettabili in risposta ad



Struttura del documento

1. Introduzione

- Descrive le funzionalità generali del sistema

2. Glossario

- Definisce tutti i termini tecnici usati nel documento.
Non deve esserci nessun prerequisito richiesto al lettore.

3. Modelli del sistema

- Mostra i modelli del sistema, evidenziando le componenti del sistema e le relazioni tra esse e con l'ambiente.

4. Definizione dei requisiti funzionali

- I servizi che devono essere offerti



5. Definizione dei requisiti non funzionali

- I vincoli sul sistema e sul processo di sviluppo

6. Evoluzione del sistema

- Le assunzioni fondamentali su cui il sistema si basa, e i possibili cambiamenti in risposta all'evoluzione dell'hardware...

7. Specifica dei requisiti

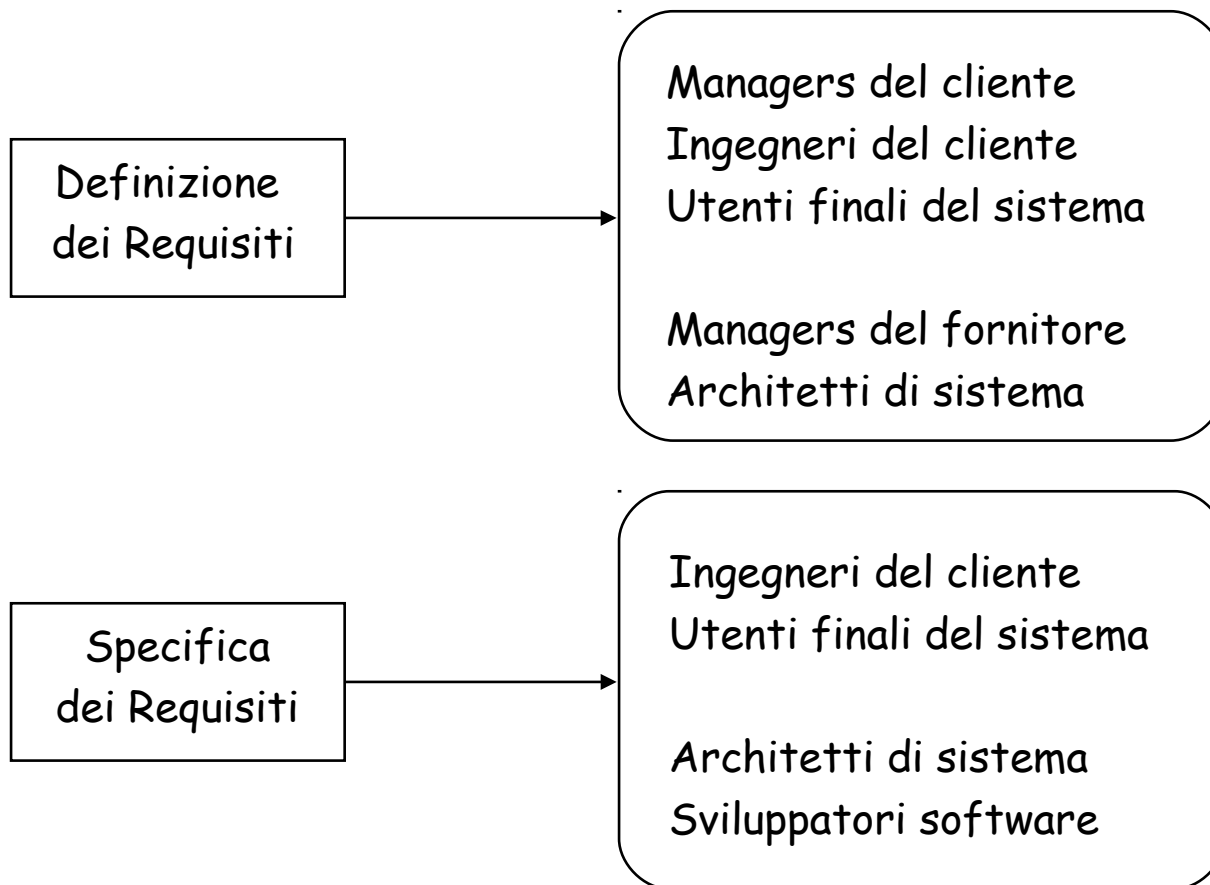
- Specifica dettagliata dei requisiti funzionali

8. Appendici

- Descrizione della piattaforma hardware
- Requisiti sui database: organizzazione logica dei dati usati dal sistema e interdipendenza tra dati

- Indice

Destinatari delle varie parti del documento di specifica requisiti





Validazione dei requisiti

- E' essenziale dimostrare che i requisiti definiscono proprio il sistema che l'acquirente desidera
- Errori nella specifica dei requisiti sono estremamente costosi
 - correggere un errore nella specifica dei requisiti a sviluppo ultimato può costare 100 volte più che fissare un errore d'implementazione
- Prototyping come tecnica di validazione



Verifica dei requisiti

- Validità.
 - Il sistema offre le funzioni che meglio rispondono alle necessità del cliente?
- Consistenza.
 - Ci sono conflitti tra i requisiti?
- Completezza.
 - Sono incluse tutte le funzioni richieste dall'acquirente?
- Realizzabilità.
 - I requisiti possono essere implementati, dato il budget e la tecnologia disponibili?



Verifica dei requisiti (2)

- Verificabilità.
 - I requisiti sono testabili realisticamente?
- Comprensibilità.
 - I requisiti sono specificati in ogni dettaglio?
- Tracciabilità.
 - E' chiara l'origine di ogni requisito?
- Adattabilità.
 - E' possibile modificare un requisito senza rimetterne in discussione gli altri?

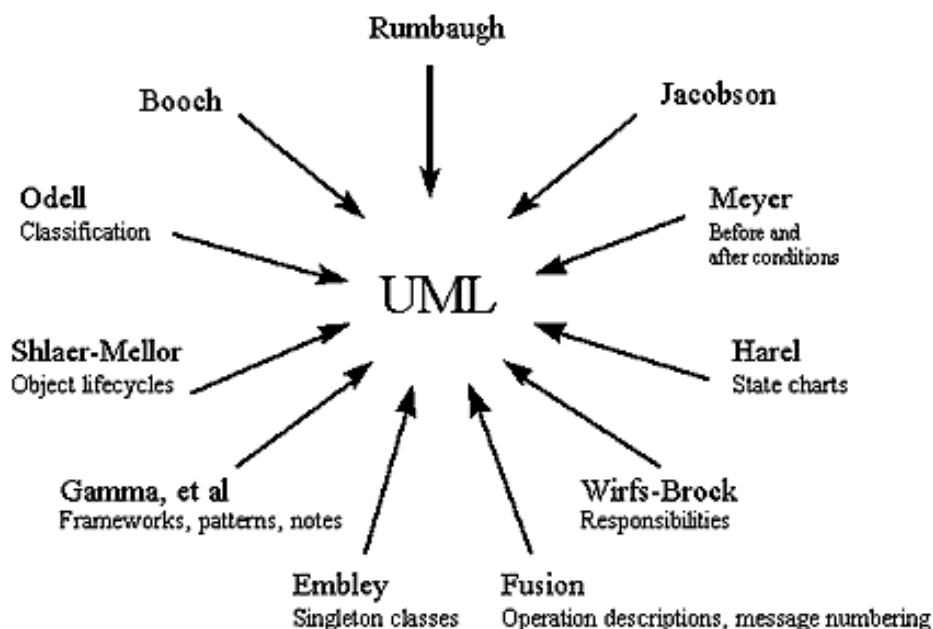


Modelli di Sistema

- Modello = Presentazione astratta del sistema di cui si stanno analizzando i requisiti, non una rappresentazione alternativa
- Serve a capire la funzionalità del sistema e a comunicare con il cliente

Unified Modeling Language

- è l'unificazione dei metodi:
 - Booch-93 di **Grady Booch**
 - OMT di **Jim Rumbaugh**
 - OOSE di **Ivar Jacobson**



- ha accolto inoltre le idee di numerosi altri metodologie
- è tuttavia indipendente dai metodi, dalle tecnologie, dai produttori



UML - linguaggio universale

- **linguaggio** per specificare, costruire, visualizzare e documentare gli artefatti di un sistema
- **universale**: può rappresentare sistemi molto diversi, da quelli web ai legacy, dalle tradizionali applicazioni Cobol a quelle object oriented e a componenti
- è un **linguaggio di modellazione**, non un metodo
- definisce una notazione standard, basata su un meta-modello integrato degli “elementi” che compongono un sistema software



Università
Ca' Foscari
Venezia

Diagrammi UML

livello “logico”:

dei casi d'uso

Use Case Diagram

delle classi

Class Diagram

di sequenza

Sequence Diagram

di collaborazione

Collaboration Diagram

di transizione di stato

Statechart Diagram

delle attività

Activity Diagram

livello “fisico”:

dei componenti

Component Diagram

di distribuzione dei componenti

Deployment Diagram



Università
Ca' Foscari
Venezia

Diagramma dei casi d'uso

Mostra:

- le modalità di utilizzo del sistema (casi d'uso)
- gli utilizzatori e coloro che interagiscono con il sistema (attori)
- le relazioni tra attori e casi d'uso

Un **caso d'uso**

- rappresenta un possibile “modo” di utilizzo del sistema
- descrive l'interazione tra attori e sistema, non la “logica interna” della funzione

una funzionalità dal punto di vista di chi la utilizza



Università
Ca' Foscari
Venezia

Perché costruire casi d'uso

- 1. Esplicitare e comunicare a tutti gli stakeholder i requisiti funzionali del sistema**– In generale, analizzare, identificare, descrivere gli usi tipici del sistema da parte dei suoi utilizzatori. Considerare anche tutta l'eventuale logica derivante dalla gestione di errori, eccezioni, flussi alternativi.
- 2. Validare i requisiti utente** – Essendo il modello dei casi d'uso piuttosto semplice (pochi costrutti, semantica precisa, ma intuitiva), diventa un valido strumento per assicurarci di aver sviluppato un sistema software corrispondente alle vere necessita' dell'utente.



Cosa è un use case

- Un use case cattura chi (attori) fa cosa (interazione) con il sistema. Con quale scopo (goal), senza considerare ciò che è dentro il sistema
- Un use case
 - Raggiunge un task di interesse per un attore singolo, discreto, completo, significativo, e ben definito
 - È un pattern di comportamento tra alcuni attori ed il sistema — una collezione di possibili scenari
 - È scritto usando il vocabolario del dominio applicativo
 - Definisce scopo ed intento (non azioni concrete)
 - È generale e indipendente dalla tecnologia



Use case diagram e requisiti

- I casi d'uso servono a:
 - chiarire i requisiti del committente in termini comprensibili
 - trovare aspetti comuni (riuso)
 - individuare gli attori del sistema
 - individuare gli eventi a cui il sistema deve rispondere
- Un requisito può dare origine a più casi d'uso
- Ad un caso d'uso possono venire associati più requisiti



Casi d'uso, attori, scenari e descrizioni

- Quattro concetti fondamentali:
 1. **Caso d'uso** – rappresenta una specifica interazione tra un attore e il sistema.
In UML rappresentato mediante **un'ellisse** etichettata col nome del caso d'uso.
 2. **Attore** – rappresenta un **ruolo** che caratterizza le interazioni tra utente e sistema. L'utente non è necessariamente umano; In UML rappresentato mediante un omino.
 3. **Scenario** – sequenza di **azioni** che definisce una particolare interazione tra attore e sistema.
 4. **Descrizione** – **testo** che descrive lo scenario, l'ordine temporale della sequenza di azioni, l'eventuale trattamento degli errori, gli attori coinvolti.

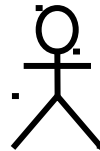


Università
Ca' Foscari
Venezia

Attori

Gli attori sono rappresentati tramite il **ruolo** che giocano nel caso d'uso e sono normalmente indicati tramite l'icona:

<<Attore>>



Cassiere

Relazione tra attori

- E' possibile definire gerarchie di attori
- Associare un attore ad uno o più attori specializzati

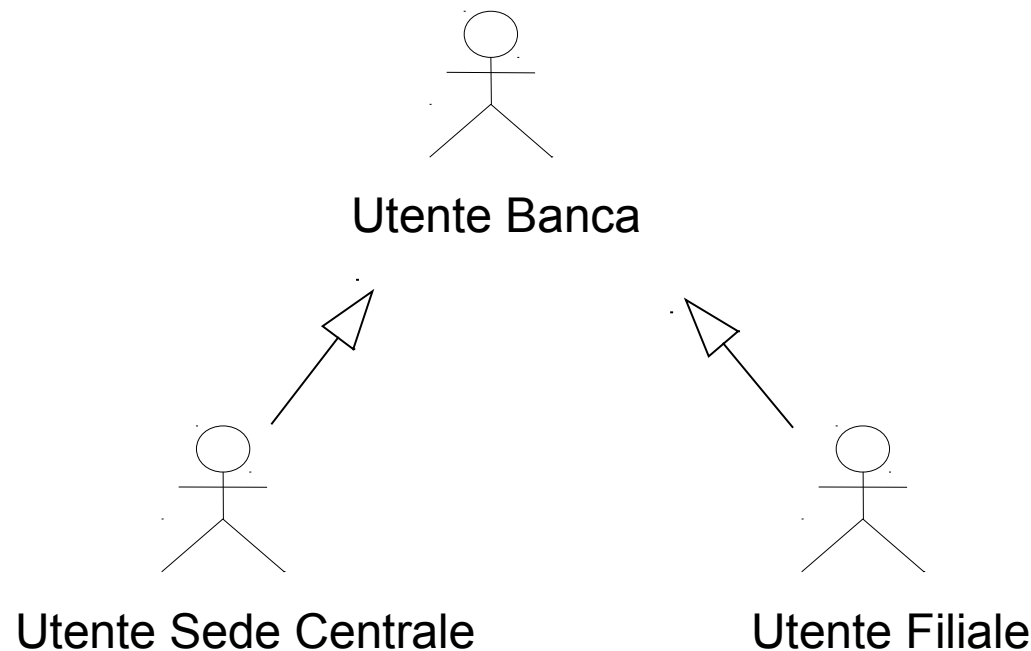
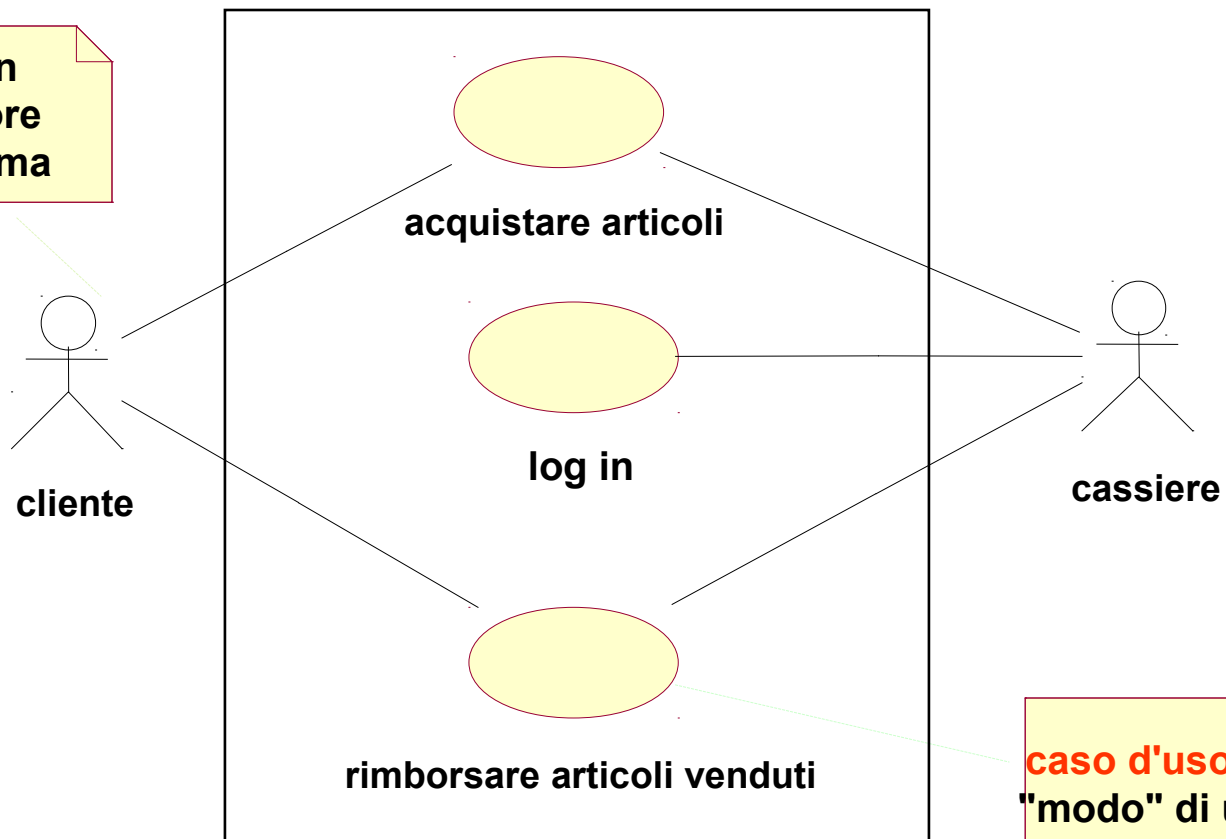


Diagramma dei casi d'uso

attore: un
utilizzatore
del sistema



caso d'uso: un
"modo" di utilizzare il
sistema

Template per la documentazione di un caso d'uso

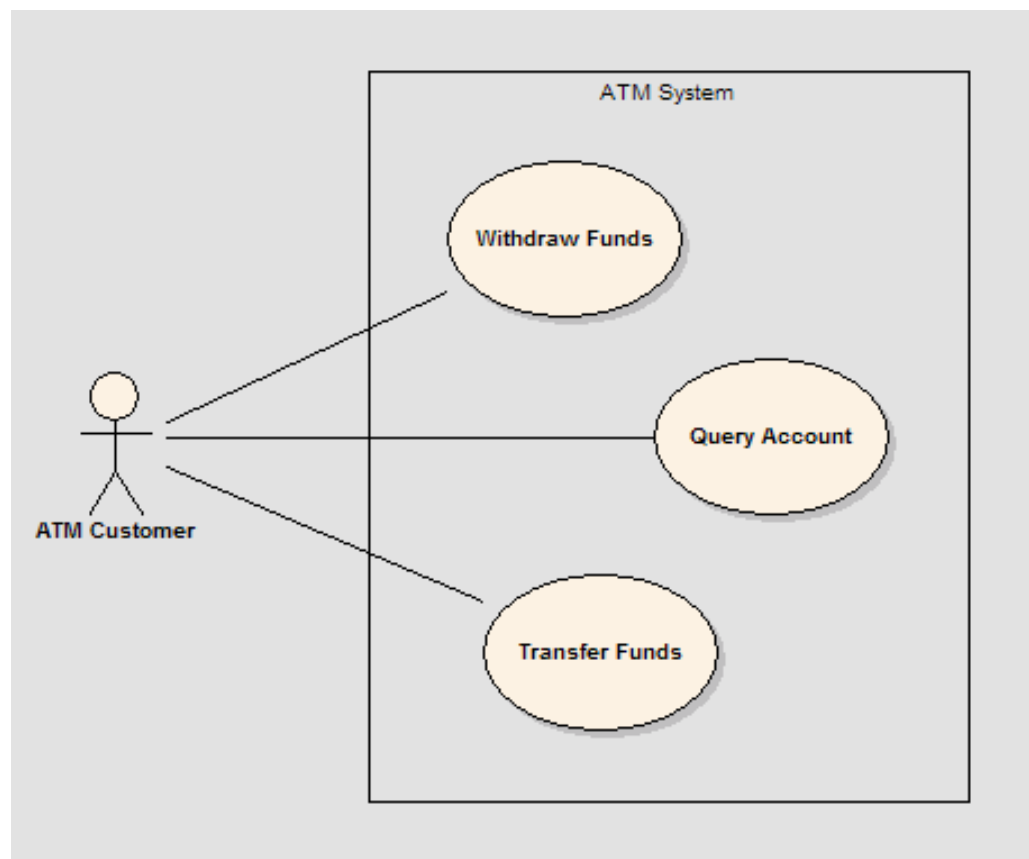
TEMPLATE DI BASE

- *Nome caso d'uso* – Ogni caso d'uso deve avere un nome; il nome esprime il goal dell'utente nell'utilizzo del sistema.
- *Goal (summary description)* – descrizione della funzionalità fornita dal sistema e che soddisfa una necessità dell'utente, ossia che è percepita dallo stesso utente come "valore".
- *Attori* – persona, dispositivo o altra entità esterna al sistema che interagisce con il sistema. Per ogni caso d'uso esiste sempre un attore primario che è colui che inizia il caso d'uso stesso.
- *Precondizioni* – Condizioni che devono essere soddisfatte all'inizio del caso d'uso. Rappresentano le "garanzie minime" che devono essere soddisfatte per poter attivare lo scenario di utilizzo del sistema (Garanzie fornite dagli attori al sistema).
- *Trigger* – evento trigger che attiva il caso d'uso
- *Descrizione (main success scenario o scenario principale)* – descrizione della sequenza di interazioni più comune tra gli attori e il sistema. In particolare viene descritta la sequenza principale che porta alla conclusione del caso d'uso con successo. La descrizione è definita in termini di input forniti dall'attore e di risposta del sistema. Il sistema è trattato secondo un modello di tipo black-box, concentrandosi su cosa esso fa in risposta agli input, e non su come internamente queste risposte vengano prodotte.
- *Alternative (estensioni)* – descrizioni delle variazioni dalla sequenza di passi tipica del main success scenario. Tali alternative estendono lo scenario principale. La gestione delle eccezioni è un esempio tipico di tali estensioni. Non tutte le alternative portano necessariamente ad un fallimento del caso d'uso.
- *Postcondizioni* – Condizioni sempre soddisfatte al termine del caso d'uso (Garanzie fornite dal sistema agli attori)



Università
Ca' Foscari
Venezia

Esempio di scenario: Prelievo dal Bancomat



Nome caso d'uso: Withdraw Funds.

Scope: ATM system

Goal (summary): L'utente dell'ATM preleva una specifica somma di denaro da un conto corrente bancario valido.

Dipendenze con altri casi d'uso: nessuna

Attori: Cliente del terminale ATM.

Trigger: inserimento di una carta di credito nel lettore del terminale ATM

Precondizioni: Il terminale dell'ATM è in attesa (idle) e visualizza un messaggio di "Benvenuto".

Descrizione (main success scenario o scenario principale):

1. Il cliente inserisce nel lettore del terminale ATM la sua carta di credito.
2. Il sistema riconosce la carta e ne legge il numero (card number).
3. Il sistema visualizza a terminale la richiesta di inserimento del PIN utente.
4. Il cliente inserisce il suo PIN.
5. Il sistema verifica che la carta non è scaduta, né è stata rubata o risulta smarrita.
6. Il sistema verifica che il PIN inserito dal cliente corrisponda con quello della carta.
7. Il sistema controlla che il conto corrente sia accessibile mediante l'utilizzo della carta.
8. Il sistema visualizza il conto corrente del cliente e visualizza le possibili operazioni che il cliente può effettuare (Prelievo, Ultimi Movimenti, Saldo Disponibile, Altri Servizi).
9. Il cliente seleziona il conto corrente desiderato (nel caso il cliente ne abbia più di uno) e seleziona l'operazione Prelievo.
10. Il sistema visualizza la richiesta dell'ammontare da prelevare.
11. Il cliente inserisce l'ammontare da prelevare.
12. Il sistema verifica che il conto corrente contenga sufficienti fondi per consentire la conclusione dell'operazione e che non sia stato superato il limite massimo giornaliero per il prelievo.
13. Il sistema autorizza il prelievo.
14. Il sistema emette il denaro, registrando la transazione sul conto corrente.
15. Il sistema stampa la ricevuta mostrando il numero della transazione, il tipo di operazione effettuata, la quantità di denaro prelevata e il saldo del conto corrente.
16. Il sistema espelle la carta.
17. Il sistema ritorna nello stato iniziale di attesa (idle), visualizzando il messaggio di "Benvenuto"

Esempio di scenario (Cont.)

Alternative (estensioni):

- 2a. Se il sistema non riconosce la carta dell'utente, quest'ultima viene espulsa dal lettore.
- 5a. Se la carta risulta scaduta, il sistema la confisca.
- 5b. Se la carta risulta smarrita, il sistema la confisca.
- 5c. Se la carta risulta rubata, il sistema la confisca.
- 6a. Se il cliente ha inserito un PIN che non corrisponde con quello della carta, il sistema visualizza una nuova richiesta di inserimento del PIN. Se il cliente inserisce per tre volte un PIN errato, il sistema confisca la carta.
- 7a. Se il sistema verifica che il numero di conto non è valido, viene visualizzato un messaggio di errore e la carta viene espulsa dal lettore.
- 12a. Se non ci sono sufficienti fondi nel conto corrente, il sistema visualizza un messaggio di errore ed espelle la carta. Il terminale ATM viene riportato allo stato di attesa (idle) e viene visualizzato il messaggio di "Benvenuto".
- 12b. Se il limite giornaliero massimo di prelievo è stato già raggiunto, il sistema visualizza un messaggio di errore ed espelle la carta. Il terminale ATM viene riportato allo stato di attesa (idle) e viene visualizzato il messaggio di "Benvenuto".
- 14a. Se il terminale ATM non ha sufficienti fondi per completare la transazione, il sistema visualizza un messaggio di errore, espelle la carta, il terminale ATM viene riportato allo stato di attesa (idle) e viene visualizzato il messaggio di "Benvenuto".
- 14b. Se il cliente seleziona da terminale il pulsante Cancel, il sistema cancella la transazione ed espelle la carta. Il terminale ATM viene riportato allo stato di attesa (idle) e viene visualizzato il messaggio di "Benvenuto".

Postcondizioni: L'utente ha ottenuto la somma di denaro che aveva richiesto di prelevare.

Questioni aperte: nessuna.



Università
Ca' Foscari
Venezia

Relazioni tra casi d'uso

- Relazione di **inclusione** «include»
- Relazione di **generalizzazione**
- Relazione di **estensione** «extend»
- Poche relazioni: modello semplice, senza grossa complessità sintattica



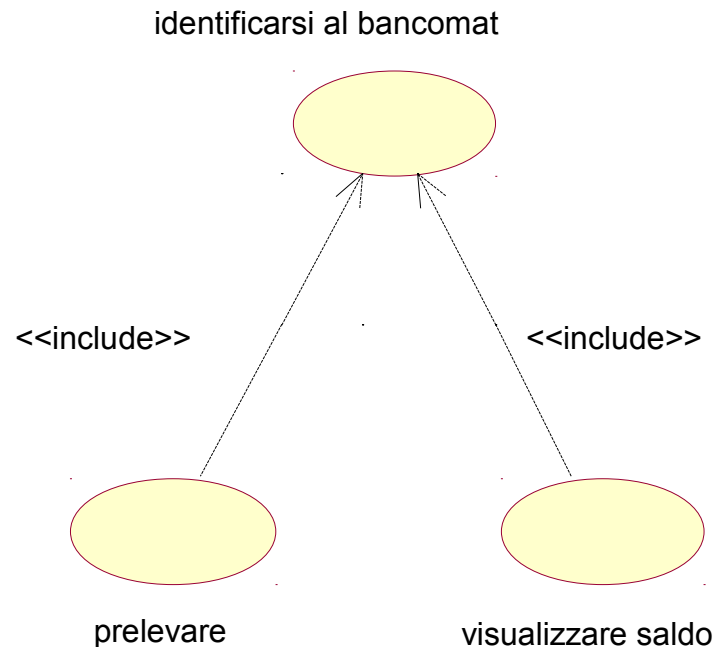
Università
Ca' Foscari
Venezia

Relazioni tra casi d'uso: inclusione

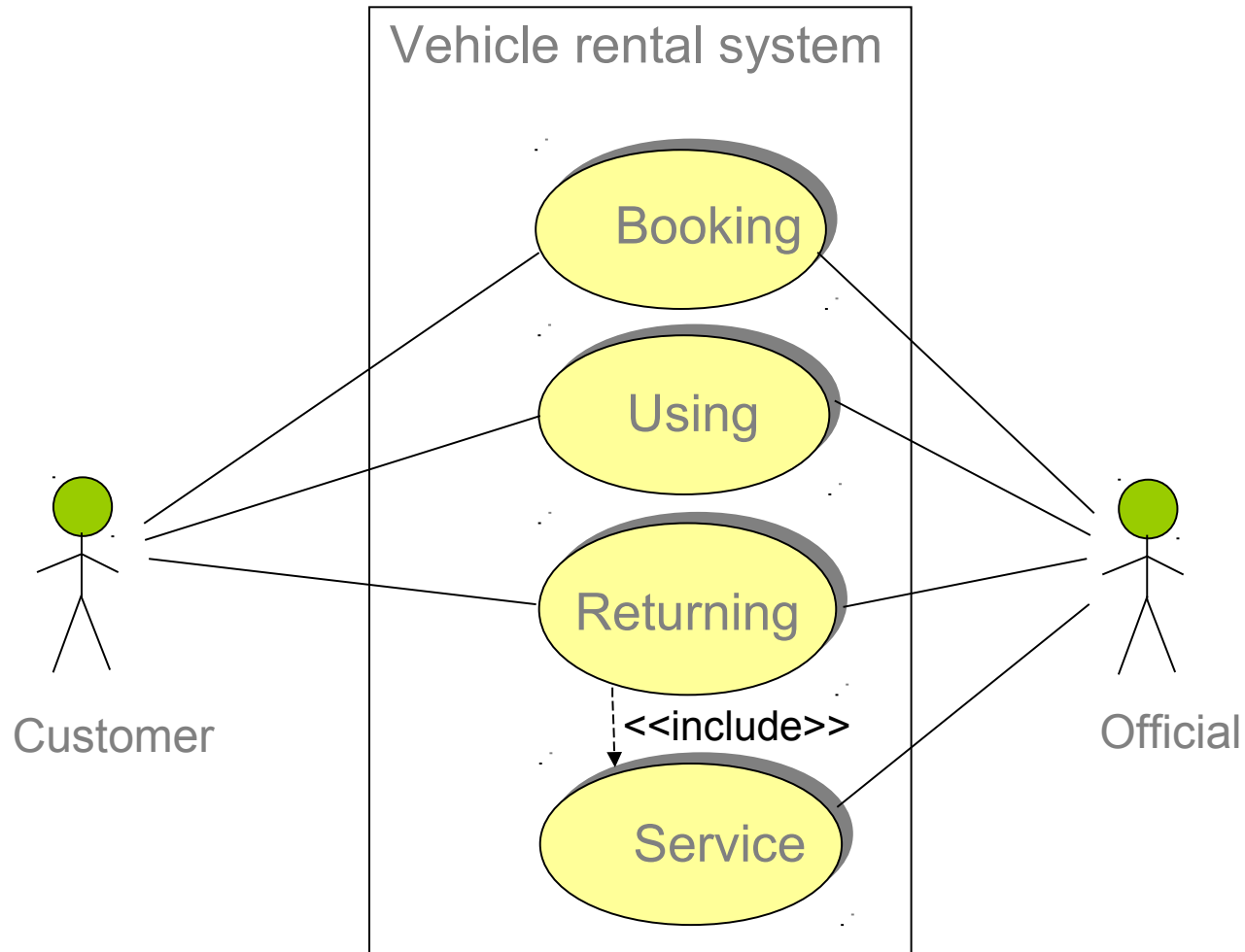
- Rappresenta l'**invocazione di un caso d'uso** da parte di un altro.
- Simile alla **chiamata di una funzione**
- Serve per scomporre casi d'uso complessi in casi d'uso più semplici
- Esprime il riuso di singoli casi d'uso

Relazioni tra casi d'uso: include

<<include>> : può mostrare anche il **comportamento comune** a uno o più casi d'uso



Esempio

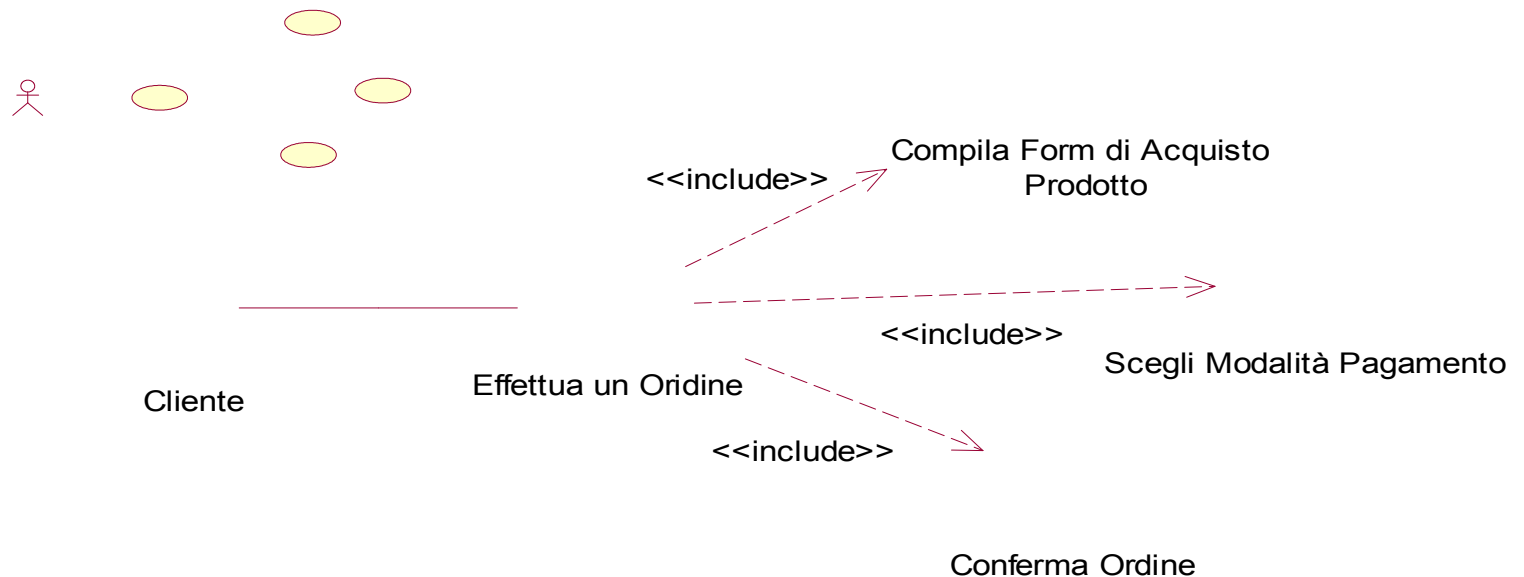




Università
Ca' Foscari
Venezia

Esempio

- Effettuare un ordine di acquisto include (sempre!) i seguenti passi:
 1. Compilazione di un form di acquisto
 2. Scelta della modalità di pagamento;
 3. Conferma dell'ordine di acquisto



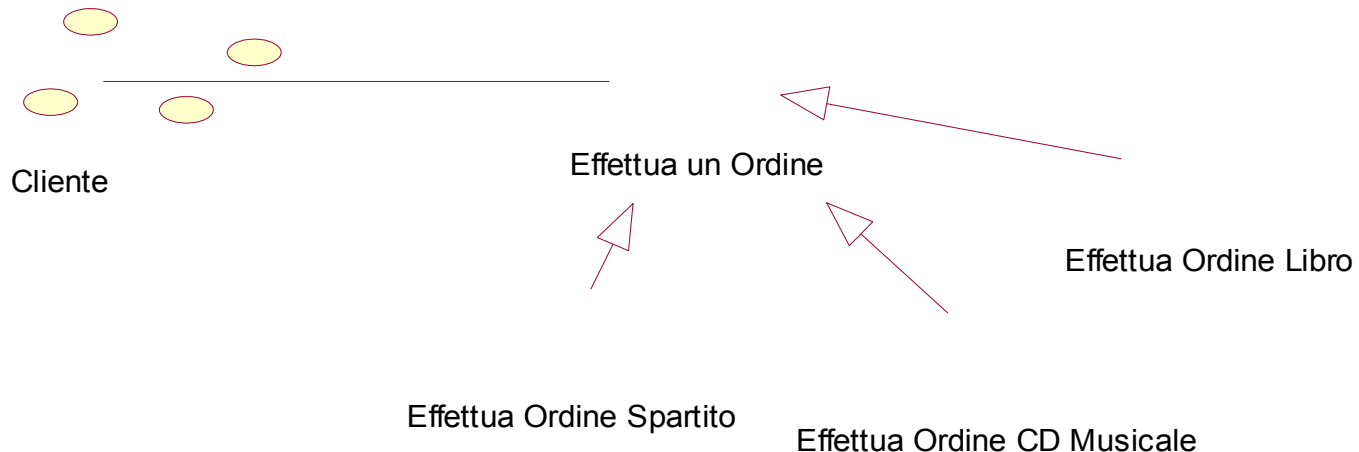


Relazioni tra casi d'uso: generalizzazione

- Simile all'ereditarietà tra classi
- Relazione che lega un caso d'uso generico a casi d'uso che sono particolari **specializzazioni** (realizzazioni) del primo
- Logica del caso d'uso derivato simile a (ma diversa da) quella del caso d'uso di base
- Viene riscritta (specializzata) la sequenza delle azioni di base oppure viene elaborata una sequenza alternativa

Relazioni tra casi d'uso: generalizzazione

- Effettua un Ordine è un caso d'uso generale che può essere specializzato nei seguenti:
 1. Effettua un ordine di un cd musicale
 2. Effettua un ordine di un libro
 3. Effettuare un ordine di uno spartito





Università
Ca' Foscari
Venezia

Relazioni tra casi d'uso: estensione

- Rappresenta l'**estensione della logica di base** di un caso d'uso.
- Il caso d'uso estensione continua il comportamento del caso d'uso di base inserendovi delle azioni alternative da un certo punto in poi (chiamato **punto di estensione**)
- Il caso d'uso di base dichiara tutti i possibili punti di estensione
- Simile alla gestione degli interrupt hardware (gestione eccezioni)

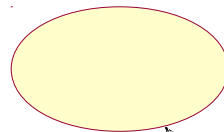


Università
Ca' Foscari
Venezia

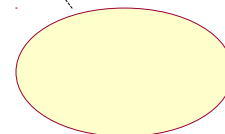
Relazioni tra casi d'uso: extend

<<extend>>: mostra il **comportamento opzionale**
(alternativo o relativo al trattamento di **condizioni anomale**)

aprire conto corrente



<<extend>>

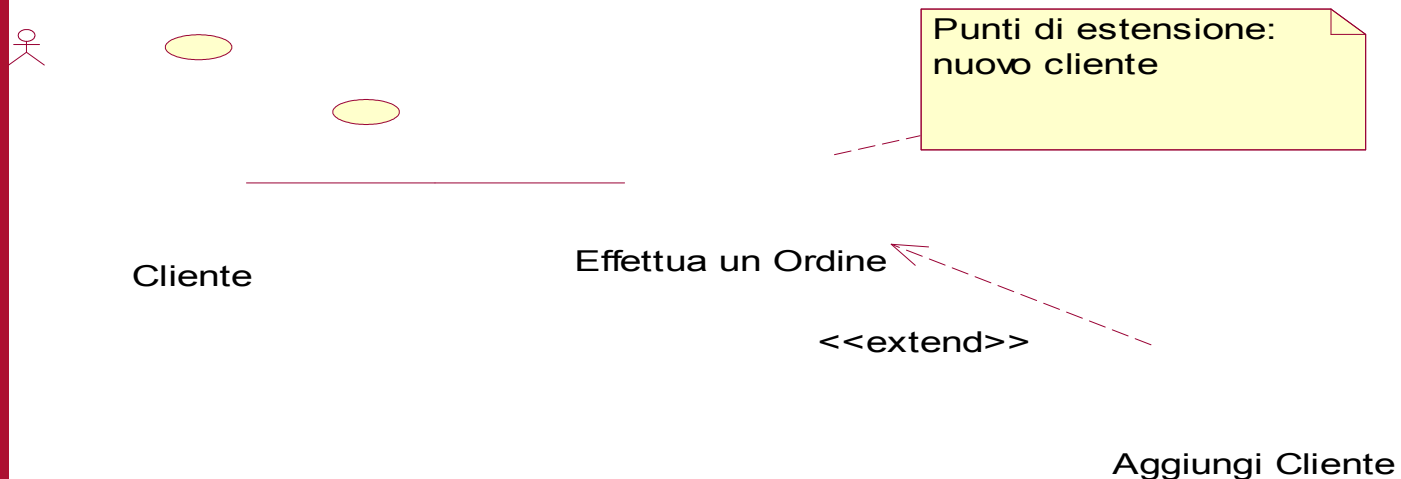


trattare condizioni particolari



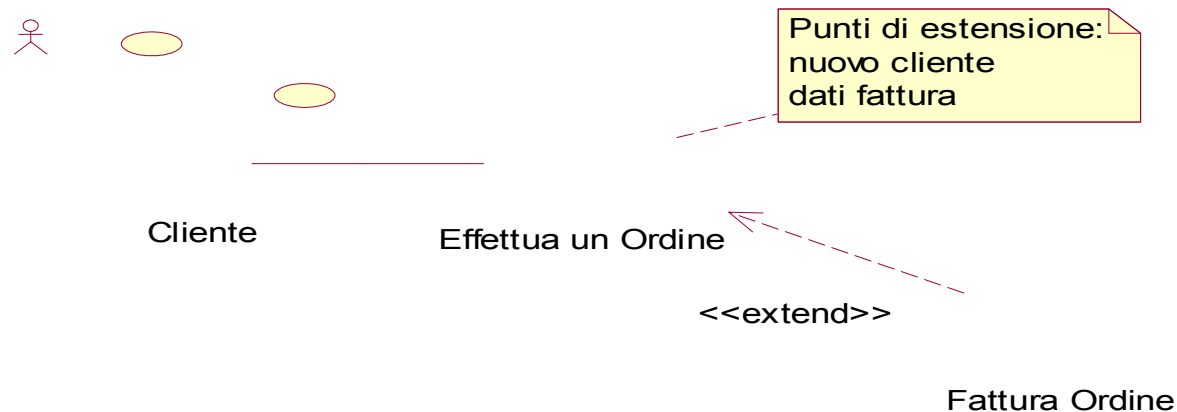
Relazioni tra casi d'uso: estensione

- Effettua un Ordine presuppone che l'utente sia già registrato nel sistema informatico
- Se il cliente è al suo primo ordine e non si è precedentemente registrato? Eccezione nel caso d'uso base!



Relazioni tra casi d'uso (estensione)

- Effettua un Ordine prevede una procedura d'acquisto di default che non invia la fattura
- E se il cliente richiede una fattura? Eccezione nel caso d'uso base



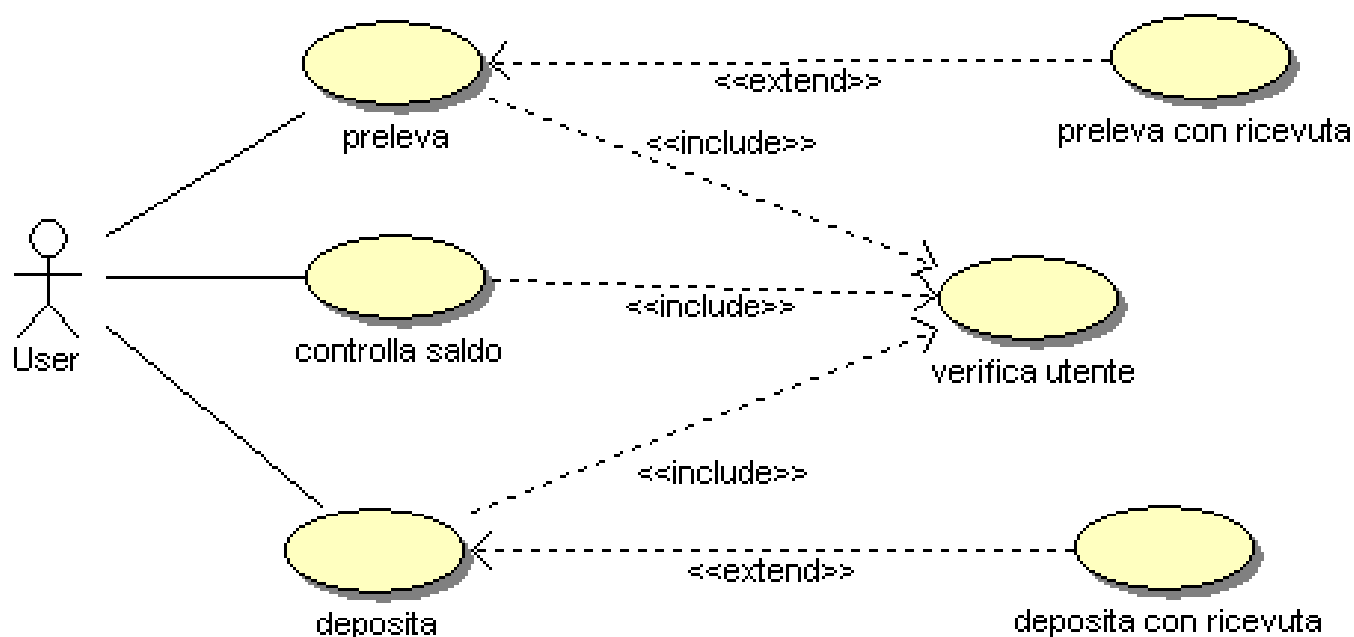


Università
Ca' Foscari
Venezia

Esercizio: Sportello Bancomat

- L'utente deve essere in grado di depositare assegni sul suo conto
- L'utente deve essere in grado di prelevare i soldi dal suo conto
- L'utente deve poter interrogare il Sistema sul saldo del suo conto
- Se lo richiede, l'utente deve poter ottenere la ricevuta per la transazione.
- I tipi di transazione sono ritiro o deposito.
- La ricevuta deve indicare la data della transazione, il numero del conto, il saldo precedente e successivo la transazione
- Dopo ogni transazione, il nuovo saldo deve essere visualizzato all'utente

Esercizio: Soluzione



Definizione dei Requisiti

- Descrizione delle funzionalità del sistema e dei vincoli operazionali, orientata al cliente
- Deve specificare il comportamento esterno del sistema: i requisiti non devono essere definiti usando un modello computazionale
- Include requisiti funzionali e non funzionali
 - **Requisiti funzionali**: dichiarazione di servizi che il sistema deve offrire
 - **Requisiti non funzionali** = vincoli sui servizi e sulle operazioni che il sistema deve realizzare



Scrivere una definizione di requisito

- Il modo usuale di scrivere le definizioni dei requisiti è usando il linguaggio naturale arricchito da diagrammi e tabelle
- Problemi:
 - Mancanza di chiarezza. La precisione è direttamente proporzionale alla difficoltà di leggere il documento.
 - Confusione tra requisiti. Requisiti funzionali e non-funzionali tendono ad essere mescolati
 - Amalgama di requisiti. Diversi requisiti espressi tutti in una volta anziché in modo distinto

Esempio

4.A.5 Il database deve supportare la creazione e il controllo di oggetti di configurazione, ovvero di oggetti che sono a loro volta raggruppamenti di altri oggetti nel database. Le facilities di controllo di configurazione devono consentire di accedere a un raggruppamento di oggetti usando un nome incompleto.

- E' un caso di amalgama: si mettono assieme concetti (prima frase) e dettagli (seconda frase)

Esempio

2.6 Griglia Per aiutarsi nel posizionamento di entità in un diagramma, l'utente può attivare una griglia, in centimetri o pollici, attraverso una opzione del pannello di controllo.

La griglia può essere attivata o disattivata in qualsiasi momento del processo di editing e può essere impostata in centimetri o pollici in qualsiasi momento. Nella riduzione delle dimensioni del documento nella finestra (reduce-to-fit) il numero di linee della griglia deve essere ridotto per evitare di riempire il diagramma solo con linee fitte della griglia.

Esempio

2.6 Griglia Per aiutarsi nel posizionamento di entità in un diagramma, l'utente può attivare una griglia, **in centimetri o pollici, attraverso una opzione del pannello di controllo.**

La griglia può essere attivata o disattivata in qualsiasi momento del processo di editing e può essere impostata in centimetri o pollici **in qualsiasi momento**. Nella riduzione delle dimensioni del documento nella finestra (reduce-to-fit) il numero di linee della griglia deve essere **ridotto** per evitare di riempire il diagramma solo con linee fitte della griglia.

- **Confusione di requisiti funzionali e non funzionali**
- **La definizione è incompleta**



Regole di stesura

- Aderire ad un formato standard: permette di evitare omissioni e semplifica i controlli incrociati
- Raggruppare i requisiti che sono legati fra loro
- Evidenziare i requisiti principali
- Associare motivazioni ai requisiti

Definizione di una griglia di un editor

2.6 Griglia

2.6.1 L'editor deve offrire una griglia di linee orizzontali e verticali come sfondo della finestra di editing.

Questa griglia deve essere una griglia passiva.

E' lasciata all'utente la responsabilità di allineare o meno le entità.

Motivazione: Una griglia aiuta l'utente a creare diagrammi ordinati con entità ben posizionate. A differenza di una griglia attiva, dove le entità sono “catturate” dalle linee della griglia, il posizionamento è impreciso. E' l'utente che deve decidere dove posizionare le entità.

2.6.2 Quando la griglia viene usata in modalità 'reduce-to-fit' (see 2.1), il numero di punti che separano le linee della griglia deve essere incrementato.

Motivazione: Se non si incrementa la spaziatura tra linee, lo sfondo diventerebbe offuscato dalle linee della griglia.



Motivazione del requisito

- La motivazione di un requisito è importante per aiutare lo sviluppatore a capire il dominio di applicazione e a capire perché il requisito è descritto in quella forma
- E' particolarmente importante quando si debbano modificare i requisiti: la presenza della motivazione di un requisito riduce la probabilità che il cambiamento produca effetti inaspettati.



3.5.1 Adding nodes to a design

3.5.1.1 The editor shall provide a facility where users can add nodes of a specified type to a design. Nodes are selected (see 3.4) when they are added to the design.

3.5.1.2 The sequence of actions to add a node should be as follows:

1. The user should select the type of node to be added.
2. The user moves the cursor to the approximate node position in the diagram and indicates that the node symbol should be added at that point.
3. The symbol may then be dragged to its final position.

Rationale: The user is the best person to decide where to position a node on the diagram. This approach gives the user direct control over node type selection and positioning.

Specification: ECLIPSE/WS/Tools/DE/FS. Section 3.5.1



Requisiti non funzionali

- Proprietà di comportamento del sistema: affidabilità, tempi di risposta e di memorizzazione, vicoli sull'I/O, ecc.
- Possono essere più critici dei requisiti funzionali, perché possono rendere inutile il sistema se non sono soddisfatti
- Tipologia dei requisiti non funzionali
 - requisiti di prodotto
 - requisiti di processo
 - requisiti esterni

Classificazione dei requisiti non funzionali

- Requisiti di prodotto
 - Specificano che il prodotto deve comportarsi in un certo modo, es. velocità di esecuzione, affidabilità, ecc.
- Requisiti di processo
 - Requisiti che sono conseguenza di scelte di tipo organizzativo, ad es. standard di processo utilizzato, requisiti sull'implementazione ecc.
- Requisiti esterni
 - Requisiti che derivano da fattori esterni al sistema e al suo processo di sviluppo, ad es. requisiti legislativi, etici ecc.



Requisiti non funzionali

- Requisiti di **Prodotto**
 - Requisiti di usabilità
 - Requisiti di efficienza
 - ➔ Performance
 - ➔ Requisiti di spazio
 - Requisiti di affidabilità
 - Requisiti di portabilità
- Requisiti di **Processo**
 - Requisiti sulle deliveries
 - Requisiti sull'implementazione
 - Requisiti sugli standards
- Requisiti **Esterni**
 - ✓ Requisiti di interoperabilità
 - ✓ Requisiti etici
 - ✓ Requisiti legali
 - ➔ Privacy
 - ➔ Sicurezza

Esempio

- Requisiti di prodotto
 - **4.C.8** It shall be possible for all necessary communication between the APSE and the user to be expressed in the standard Ada character set.
- Requisiti di processo
 - **9.3.2** The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.
- Requisiti esterni
 - **7.6.5** The system shall provide facilities that allow any user to check if personal data is maintained on the system. A procedure must be defined and supported in the software that will allow users to inspect personal data and to correct any errors in that data.



Verificabilità dei requisiti

- I requisiti devono essere scritti in modo da poter essere facilmente verificati
- Evitare termini vaghi: non ha senso scrivere
 - Il sistema deve essere facile da usare per un operatore specializzato e dev'essere organizzato in modo da minimizzare gli errori
- Ma bisogna “quantificare”:
 - Un operatore specializzato deve essere in grado di utilizzare il sistema dopo due ore di training. Dopo tale training, il numero medio di errori fatti da un operatore specializzato deve essere inferiore a due per giorno lavorativo



Università
Ca' Foscari
Venezia

Misure per i requisiti

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	K Bytes Number of RAM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems



Requisiti a livello di sistema

- Alcuni requisiti pongono dei vincoli al sistema nella sua interezza, non a funzioni specifiche
- Esempio
 - Il tempo richiesto per la formazione degli operatori del sistema non dovrà eccedere i 2 giorni lavorativi
- Ci possono essere dei requisiti che non possono essere derivati da nessun singolo sottoinsieme di requisiti del sistema



Specifica dei Requisiti

- Descrizione precisa e dettagliata delle **funzionalità** del sistema. Serve come base da una parte per il contratto, e dall'altra per la progettazione e lo sviluppo.
- Deve essere consistente con la definizione.
- Di solito viene presentata con il modello di sistema sviluppato nell'analisi dei requisiti
- Utilizzo di form strutturate (vedi analisi dei requisiti)

Specifiche basate su “forms”

- Forma limitata di linguaggio naturale: impone un grado di uniformità alla specifica.
- **Metodo:**
 - Descrizione della funzione o entità da specificare
 - Descrizione degli input e da dove provengono
 - Descrizione degli output e dove vanno a finire
 - Indicazioni di altre entità richieste
 - Pre e post condizioni
 - Effetti collaterali (se ce ne sono)



Università
Ca' Foscari
Venezia

Specifica basata su “forms” dell’aggiunta di un nodo

ECLIPSE/Workstation/Tools/DE/FS/3.5.1

Function Add node

Description Adds a node to an existing design. The user selects the type of node, and its position. When added to the design, the node becomes the current selection. The user chooses the node position by moving the cursor to the area where the node is added.

Inputs Node type, Node position, Design identifier.

Source Node type and Node position are input by the user, Design identifier from the database.

Outputs Design identifier.

Destination The design database. The design is committed to the database on completion of the operation.

Requires Design graph rooted at input design identifier.

Pre-condition The design is open and displayed on the user's screen.

Post-condition The design is unchanged apart from the addition of a node of the specified type at the given position.

Side-effects None

Definition: ECLIPSE/Workstation/Tools/DE/RD/3.5.1



Tracciabilità dei requisiti

- Tracciabilità = requisiti correlati sono in qualche modo raggiungibili l'uno dall'altro.
- Assegna ad ogni requisito un numero (unico)
- Inserisci riferimenti incrociati a requisiti che sono correlati
- Scrivi una matrice di riferimenti incrociati che evidenzia la correlazione tra requisiti.
Occorreranno diverse matrici per i diversi tipi di relazione