Analisi dei requisiti

Alberto Gianoli

Univ. Ferrara - Corso di Laurea in Informatica

Dove?

- Pressman, cap. 10
- Sommerville, cap. 7

Definizione e specifica

- Tecniche per analizzare, definire e specificare i requisiti dei sistemi software
- Lo scopo è capire meglio il problema che si sta cercando di risolvere: capirne l'impatto, ciò che vogliono i clienti, come lo useranno gli utenti finali
- Varie fasi: avvio, raccolta, elaborazione. In più la negoziazione: priorità, cosa è fondamentale, tempi...
- Il prodotto finale è una documentazione scritta del problema

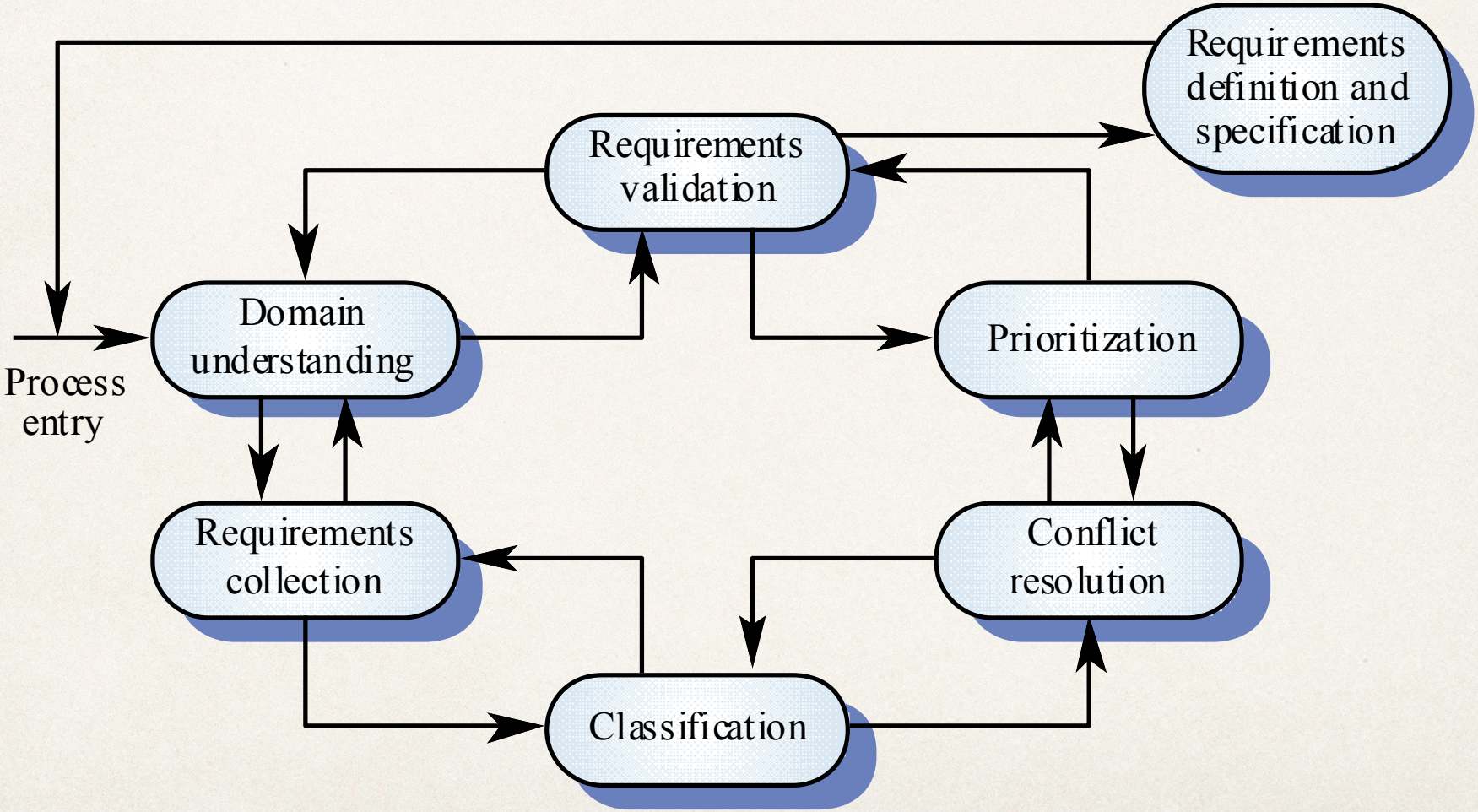
L'analisi dei requisiti

- ***Coinvolge sicuramente*** lo staff tecnico: collaborano con il committente per individuare il dominio applicativo, i servizi che il sistema deve offrire e i vincoli operativi
- Sono ***quasi certamente coinvolti*** anche utenti finali, manager, ingegneri coinvolti nella manutenzione, esperti del dominio, ecc.
  - queste sono gli *stakeholders (controparti)*

Problemi classici dell'analisi dei requisiti

- Gli stakeholders non hanno le idee chiare sul cosa vogliono
- Gli stakeholders esprimono i requisiti usando una loro terminologia (di solito la terminologia è legata al dominio applicativo)
- Tra gli stakeholders ci possono essere requisiti contrastanti
- Possono esserci altri fattori (politica interna, gestione) che influenzano i requisiti del sistema
  - ricordarsi che i requisiti possono cambiare e che possono saltar fuori nuovi stakeholders

Il processo di analisi dei requisiti



Requisiti vs progettazione e costruzione

- Progettare e costruire sw può essere difficile, richiede anche creatività e può essere divertente.
- E' coinvolgente: spesso si inizia a realizzare sw prima di avere una idea chiara. Il presupposto è che le cose diventeranno più chiare andando avanti: stakeholders comprenderanno meglio i bisogni esaminando le prime iterazioni del sw; quello che conta è produrre sw funzionante e tutto il resto è secondario; le cose cambiano rapidamente e l'ingegneria dei requisiti fa perdere tempo. C'è una parte di verità e può condurre in errore.
- Ing. requisiti deve essere adattata ai bisogni del processo, del progetto, del prodotto e delle persone coinvolte
- In alcuni casi si può abbreviare, in altri conviene sia rigorosa: il team di sviluppo deve adattare il proprio approccio all'ing. requisiti
  - però il team deve realmente comprendere i requisiti di un problema *prima* di tentare di risolverlo

Attività

- Comprensione del dominio
  - gli analisti devono “padroneggiare” il dominio applicativo: se il sistema riguarda la gestione di una carrozzeria, gli analisti devono diventare “esperti” nelle varie attività di gestione di una carrozzeria
- Raccolta dei requisiti
  - interagire con gli stakeholders per identificare i loro bisogni (e quindi i requisiti)
- Classificazione
  - prendere l’insieme non strutturato dei requisiti e organizzarli in modo coerente

Attività

- Risoluzione dei conflitti
  - a causa del numero di stakeholders coinvolti, possono esserci dei conflitti nei requisiti: vanno individuati e risolti
- Assegnazione delle priorità
  - non tutti i requisiti sono uguali: alcuni sono più importanti di altri; bisogna stabilire delle classi di priorità
- Validazione dei requisiti
  - i requisiti vanno ricontrollati per vedere se sono completi, consistenti e in accordo con quello che i stakeholders vogliono realmente dal sistema

Avvio

- In quale modo parte un progetto sw? Basta anche una conversazione al bar
- In generale si inizia perchè viene identificato un bisogno, un mercato potenziale, un servizio.
- La componente business degli stakeholders (business manager, personale marketing, manager con responsabilità sui prodotti, ..) definiscono:
  - un business case per l'idea (analizzano fattori di business a supporto)
  - tentano di identificare ampiezza e profondità del mercato
  - primo studio approssimativo di fattibilità
  - identificano una descrizione dello scope del progetto
- Queste informazioni sono soggette a cambiamenti, ma permettono di avviare discussioni con chi si occupa di ingegneria del software

Avvio - stakeholders

- Stakeholder: chiunque tragga benefici diretti o indiretti dal sistema che viene sviluppato
  - manager che si occupano delle attività commerciali
  - manager che si occupano del prodotto
  - personale marketing
  - clienti interni/esterni
  - utenti finali
  - consulenti
  - ingegneri sw
  - addetti supporto e manutenzione
  - .....
- Ognuno di loro ha un punto di vista differente, avrà benefici differenti dal sistema, avrà rischi differenti se il progetto non va in porto

Avvio - stakeholders

- I responsabili della raccolta dei requisiti devono farsi un elenco delle persone a cui sarà richiesto di collaborare alla definizione dei requisiti
  - l'elenco cresce mano a mano che si parlerà con loro: ognuno dovrà rispondere alla domanda "con chi altro sarebbe utile parlare?"

Avvio - punti di vista

- I requisiti vengono necessariamente esplorati da più punti di vista
  - marketing è interessato a funzionalità e caratteristiche che potranno interessare il mercato potenziale: serve per venderlo
  - manager è interessato a rientrare nel budget e a rispondere a specifiche finestre di mercato
  - utente finale è interessato a funzionalità che ritiene familiari o facili da apprendere/utilizzare
- Ogni stakeholder contribuirà con il proprio punto di vista.
- Potranno esserci requisiti incoerenti o in conflitto tra loro.
- Lo scopo, in questa fase, è catalogare le informazioni raccolte (incoerenze comprese), non decidere.

Avvio - collaborazione

- Come cercare di ottenere collaborazione da tutti?
- Cominciamo con identificare aree comuni (requisiti su cui tutti concordano) e aree di conflitto o contenenti incoerenze
- Per queste ultime un modo di procedere consiste nei “punti di priorità”
  - ogni stakeholder ha un certo numero di punti di priorità che può spendere nei vari requisiti
  - i punti spesi non possono essere riutilizzati
  - in questo modo si ottiene una indicazione dell’importanza globale di ciascun requisito secondo i vari punti di vista
- N.B. non vuol dire che i requisiti vanno per forza definiti collegialmente. In molti casi la decisione finale spetta a un direttore generale di progetto

Avvio - prime domande

- Lo scopo è definire una solida conoscenza del problema, di chi vuole una soluzione, della natura della soluzione desiderata, e delle comunicazioni/collaborazioni tra cliente e sviluppatore
- Dovrebbero essere domande aperte: non devono influenzare la risposta
  - p.e. chiedere “quali funzionalità pensi siano utili” piuttosto che presentare una lista di funzionalità e chiedere se sono utili: in questo modo lo stakeholder istintivamente si limita a prendere in considerazione quelle della lista
- Per esempi di domande, vedi la lezione sui requisiti
- N.B. la sessione domande/risposte va bene per il primo incontro: poi conviene passare a una combinazione di problem solving, negoziazione e specifica

Raccolta

- ***Sembra*** semplice: basta chiedere al cliente, utenti, .. quali sono gli obiettivi del prodotto, cosa si vuole ottenere, in quale modo il prodotto risponde ai bisogni stabiliti e come verrà utilizzato.
  - problemi di scope: limiti del sistema mal definiti o cliente specifica dettagli tecnici non necessari che confondono invece di chiarire gli obiettivi generali del sistema
  - problemi di comprensione: cliente non ben sicuro di cosa serve, ha idea vaga delle limitazioni, non perfetta conoscenza del dominio del problema, problemi a comunicare i bisogni, omette cose "ovvie", requisiti ambigui o non verificabili, ...
  - problemi di volatilità: i requisiti cambiano nel tempo
- Bisogna affrontare l'attività di raccolta dei requisiti in modo organizzato

Raccolta collaborativa

- L'approccio più usato è il Facilitated Application Specification Technique (FAST): l'abbiamo già visto quando abbiamo parlato di requisiti.
- Durante l'avvio si è definito lo scope del problema e la percezione globale di una soluzione, e gli stakeholders hanno preparato una "richiesta di prodotto".
- Prima delle riunioni, i partecipanti devono preparare (richiesta di prodotto alla mano) elenco di oggetti che fanno parte del sistema, elenco di oggetti che devono essere prodotti dal sistema, elenco di oggetti che vengono utilizzati dal sistema per le proprie funzioni; in più elenco dei servizi che manipolano o interagiscono con gli oggetti, e dei vincoli
  - elenco non esaustivo, ma deve riflettere la sua percezione del sistema

Raccolta collaborativa

- A questo punto si presentano i singoli elenchi: ogni elemento dovrebbe poter essere manipolato separatamente. No dibattiti e critiche in questa fase
- Creare un elenco combinato: si eliminano i doppioni.
- L'obiettivo è arrivare ad un elenco consensuale per ciascun argomenti: oggetti, servizi, vincoli, prestazioni.
- A questo punto si può procedere sviluppando delle "mini-specifiche", oppure sviluppando dei casi d'uso dell'utente.
- In generale ogni partecipante crea anche un elenco di criteri di validazione. Anche per i criteri di validazione serve crerare un elenco consensuale

Definizione e specifica dei requisiti

- Definizione dei requisiti
  - descrizione delle funzionalità del sistema e dei vincoli operativi, orientata al cliente
- Specifica dei requisiti
  - descrizione precisa e dettagliata delle funzionalità e dei vincoli del sistema. Serve per comunicare agli sviluppatori cosa il sistema deve fare
  - la specifica viene utilizzata come base per il contratto di sviluppo del sistema

Definizione dei requisiti

- Deve specificare il comportamento che il sistema manifesta verso l'esterno (p.e. verso gli utenti) senza definire un modello computazionale
- Include requisiti funzionali e non funzionali
  - funzionali: descrivono le funzionalità che il sistema deve fornire
  - non funzionali: vincoli sui servizi e funzionalità fornite, o sul processo di sviluppo o sull'ambiente esterno; o proprietà che il sistema non deve avere

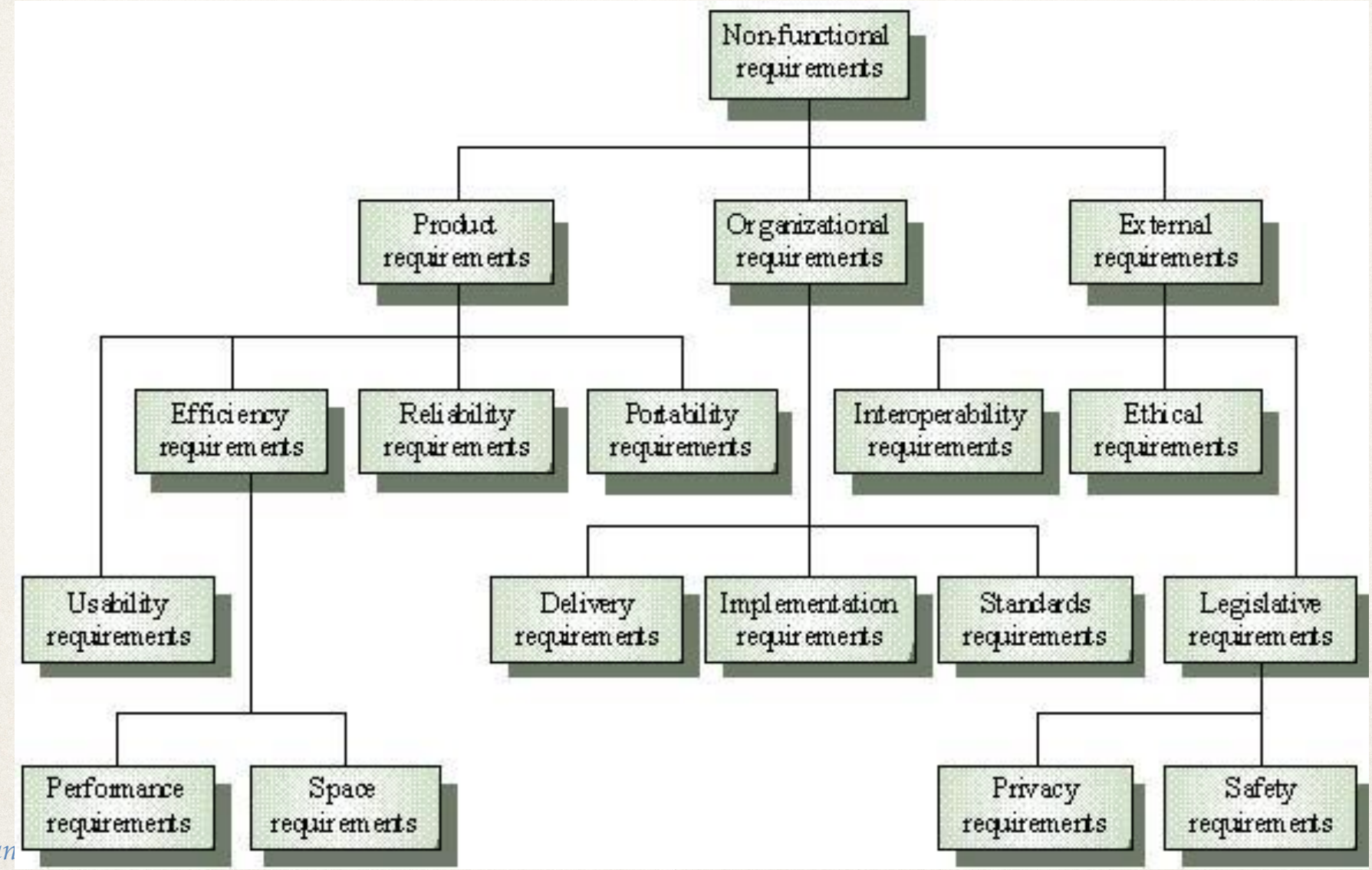
Requisiti non funzionali

- Proprietà di comportamento del sistema: affidabilità, tempi di risposta, vincoli sull'I/O, ...
- Possono essere più critici dei requisiti funzionali, perché possono rendere inutile il sistema se non vengono 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; p.e. velocità di esecuzione, affidabilità, ...
- **Requisiti di processo**
  - sono conseguenza di scelte di tipo organizzativo; p.e. standardi di processo utilizzato, requisiti sull'implementazione, ...
- **Requisiti esterni**
  - derivano da fattori esterni al sistema e al suo processo di sviluppo; p.e. requisiti legislativi, etici, ...

Requisiti non funzionali



Esempio requisiti non funzionali

- Requisiti di prodotto
  - 4.C.8 it shall be possible for all necessary communication between the APSE and the user to be espressed 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.

Come si scrive la definizione dei requisiti

- Normalmente in linguaggio naturale usando anche tabelle o diagrammi se necessario. Modo informale
- Questo modo è universalmente comprensibile ma può creare tre tipi di problemi
  - **mancanza di chiarezza**: è difficile scrivere con precisione senza rendere un documento illeggibile
  - **confusione dei requisiti**: requisiti funzionali e non funzionali tendono ad essere mescolati tra loro
  - **amalgama dei requisiti**: requisiti diversi potrebbero essere descritti insieme (p.e. nella stessa frase)

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: abbiamo mischiato 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.

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

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

- Aderite ad un formato standard: permette di evitare omissioni e semplifica i controlli incrociati
- Raggruppate i requisiti che sono legati tra loro
- Evidenziate i requisiti principali
- Associate motivazioni ai requisiti

Esempio

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 del requisito serve per aiutare lo sviluppatore a capire il dominio di applicazione e a capire perché il requisito è descritto in quella forma
- E' particolarmente importante qualora si debbano modificare i requisiti: la presenza della motivazione di un requisito riduce la probabilità che il cambiamento alteri l'intenzione originaria e che si producano effetti inaspettati

Esempio

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

Elaborazione

- Si prendono le informazioni ottenute e le si espandono e raffinano
- Occorre descrivere il problema in modo da creare solide basi per la progettazione. Se si oltrepassa questo punto, vuol dire che siamo già partiti a progettare il sistema
-

Negoziazione

- Vari motivi
  - requisiti in conflitto
  - come far combaciare le richieste con le risorse disponibili
- Può essere necessario valutare compromessi tra funzionalità, prestazioni e altre caratteristiche del prodotto da un lato e costi o tempi di uscita sul mercato dall'altro
- Lo scopo di questa fase è sviluppare un piano di progetto che risponda ai bisogni del cliente riflettendo i vincoli del mondo reale

Negoziazione

- Il trucco? Cercare di ottenere vantaggi per tutte le parti (non “tutti i vantaggi” chiaramente, ma abbastanza vantaggi)
- Libri sull’abilità di negoziazione si contano a decine, e non si applicano solo al campo dell’ingegneria del sw

Negoziazione

- In generale
  1. Riconoscere che non è una gara: entrambe le parti devono ritenere di aver vinto, ma entrambe devono scendere a compromessi
  2. Stabilire una strategia: decidere cosa si vuole ottenere, cosa vuole ottenere l'altra parte, e come si può fare in modo che accada
  3. Ascoltare attentamente: non formulare una risposta mentre l'altra parte sta ancora parlando; prima devi ascoltare
  4. Concentrarsi sugli interessi dell'altra parte: una posizione rigida aumenta i conflitti, non li risolve
  5. Evita di scendere sul piano personale: vogliamo risolvere il problema, non insultarci
  6. Essere creativi: se si è a un punto morto, non aver paura a uscire dai canoni
  7. Essere pronti ad impegnarsi: se si raggiunge un accordo non bisogna esitare ma impegnarsi a procedere

Specifica dei requisiti

- Con la specifica dei requisiti aggiungiamo dettagli alla definizione: ovviamente dobbiamo essere consistenti con la definizione...
- Di solito anche la specifica è scritta in linguaggio naturale, ma...
  - il linguaggio naturale assume che chi scrive e chi legge usino le stesse parole per esprimere gli stessi concetti (identità di vocabolario)
  - una specifica in linguaggio naturale può ammettere varie interpretazioni
  - è difficile trovare requisiti collegati tra loro, o il linguaggio naturale può non essere sufficiente a partizionare i requisiti

Alternative al linguaggio naturale

- Linguaggio naturale “strutturato”
  - definire forme standard o template per esprimere le specifiche
- Linguaggi di Descrizione di Programmi (PDL)
- una specie di linguaggi di programmazione: definiscono i requisiti fornendo una visione operazionale del sistema
- Requirements Specification Languages
- Notazioni grafiche (SADT)
- Specifiche formali
  - reti di petri
  - sistema Z
  - sistema B
  - logiche temporali

Gestione

- I requisiti possono cambiare; anzi, la necessità di cambiare i requisiti può essere richiesta per tutta la durata del sistema
- Bisogna poter identificare, controllare e tracciare i requisiti e i cambiamenti a mano a mano che si procede nello sviluppo del progetto
- Problema molto simile alla gestione della configurazione del software: quindi soluzione simile
  - assegna a ogni requisito un identificatore univoco
  - sviluppa delle tabelle di tracciabilità
    - tracciabilità delle funzionalità: relazioni tra requisito e funzionalità
    - tracciabilità dell'origine: origine di ogni requisito
    - tracciabilità e dipendenze: relazioni tra requisiti
    - ...

Gestione - Tracciabilità dei requisiti

- I requisiti in qualche modo associati devono essere collegabili in qualche modo
- La tracciabilità è una proprietà della specifica dei requisiti che si riflette nella facilità di trovare requisiti collegati
- Di solito la tracciabilità si ottiene numerando i requisiti e utilizzando questi numeri per inserire riferimenti incrociati o creare indici

Gestione

Aspetti specifici del sistema o del suo ambiente

Requisiti

	A01	A02	A03	A04	A05			Aii
R01			✔		✔			
R02	✔		✔					
R03	✔			✔				✔
R04		✔			✔			
R05	✔	✔		✔				✔
Rnn	✔	✔						

Validazione

- Esaminare le specifiche per garantire che tutti i requisiti software siano stati descritti in modo non ambiguo
- che siano state rilevate e corrette le incoerenze, omissioni, errori, ..
- Come? Revisione tecnica formale
- Esempio di domande da porsi
  - i requisiti sono affermati in modo chiaro? sono possibili interpretazione errate?
  - è stata identificata la fonte del requisito? la formulazione finale è stata confrontata con la fonte originale?
  - il requisito è limitato quantitativamente?
  - quali altri requisiti sono in relazione a questo? esiste una matrice di riferimenti incrociati?
  - i requisiti violano qualche vincolo del dominio del sistema?
  - ...

Validazione - Verificabilità dei requisiti

- I requisiti devono essere scritti in modo da poter essere facilmente verificati
- Evitate 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"
- Dovete "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"

Passiamo alla pratica

- Vediamo come mettere in pratica tutto questo
- Prendiamo un esempio che abbiamo già visto parlando di UML e facendo esempi di diagrammi UML: il bancomat
- Vediamo come l'esempio fatto si va a integrare con quanto detto fin'ora
- **N.B.** in realtà prima avremmo dovuto fare la raccolta dei requisiti, e durante l'analisi di questi passare agli use case che abbiamo visto e agli altri diagrammi

Analisi viewpoint-oriented

- Guardare al problema con gli occhi dei vari “attori”, di tutti quelli che sono in qualche modo coinvolti nel sistema
- View-points: produttori/consumatori di dati, utenti/realizzatori di servizi, interni/esterni al sistema, ...
- Perché è importante avere più “prospettive”?
  - raramente c’è un unico modo “corretto” di analizzare i requisiti di un sistema
  - quindi non c’è una singola vista corretta del sistema, ma tanti modi ugualmente corretti che lo rappresentano

Modello di processo VORD
(Viewpoint Oriented Requirements Definition)

- Identificazione dei viewpoints
  - individuare i viewpoint che ricevono servizi dal sistema e identificare i servizi specifici che il sistema deve offrire a ogni viewpoint
- Strutturazione dei viewpoint
  - combinare i viewpoint a gruppi, in modo gerarchico; servizi comuni sono forniti al più alto livello della gerarchia
- Documentazione dei viewpoints
  - raffinare la descrizione dei viewpoint e i servizi identificati
- Mapping viewpoint-sistema
  - definire gli oggetti che rappresentano i viewpoint (modello OO)

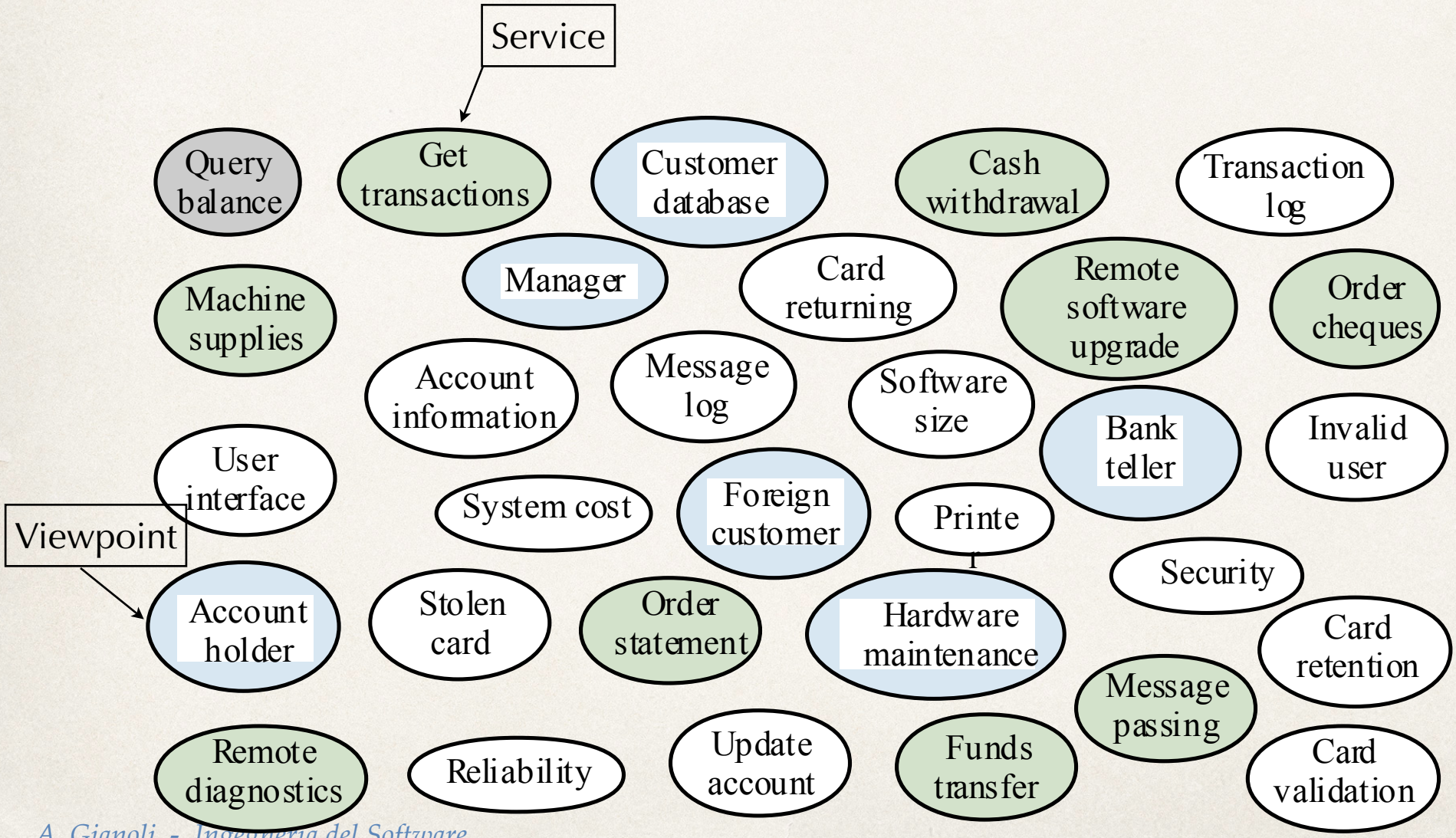
Sportello bancomat

- Sistema che offre alcuni servizi bancari automatizzati (prelievo, estratto conto, bonifico, ...)
- View-points:
  - clienti della banca
  - responsabili della manutenzione hw e sw
  - ufficio marketing
  - managers della banca e personale dello sportello
  - amministratori del database e personale sorveglianza
  - rappresentanti altre banche
  - .....

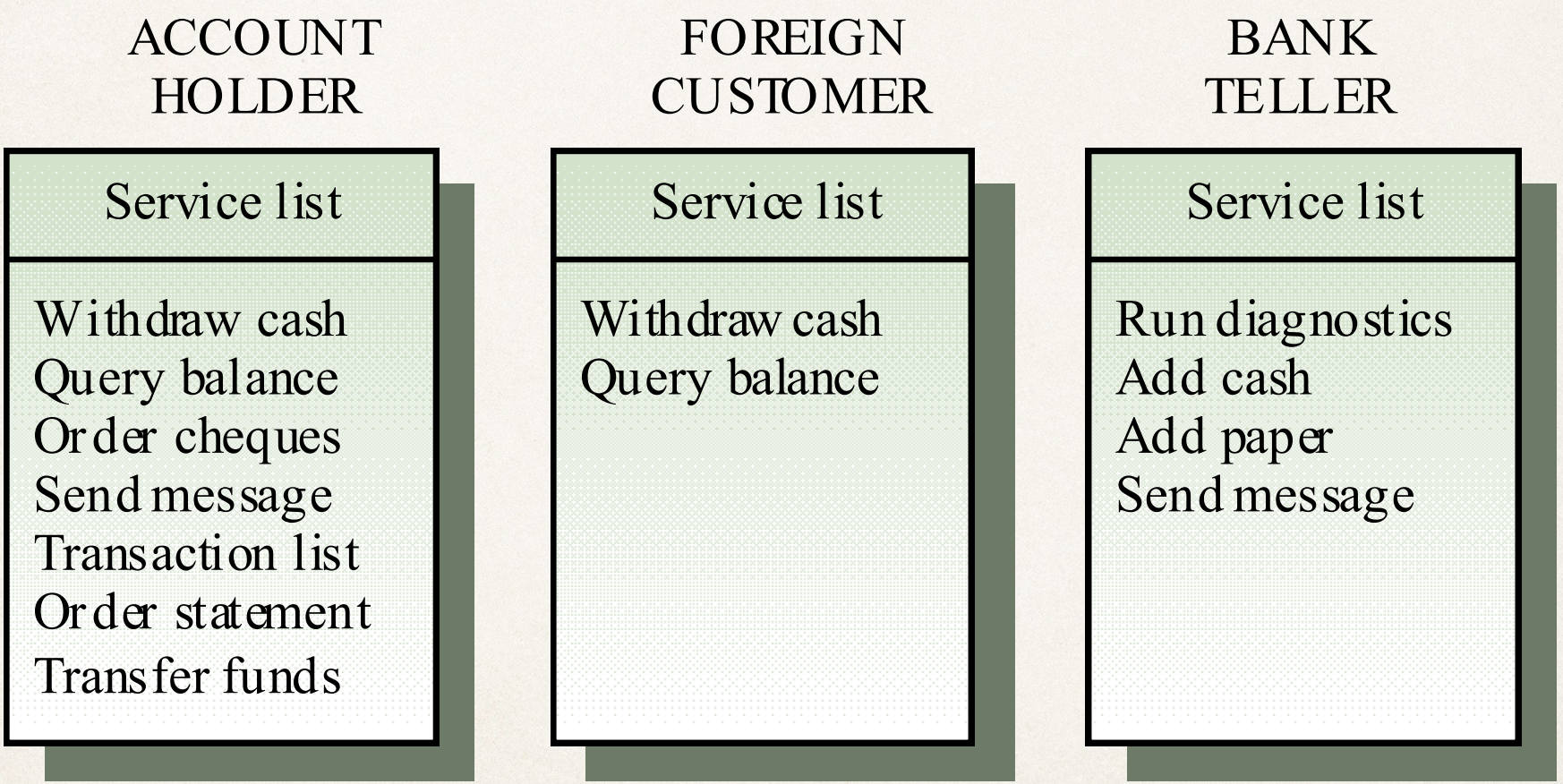
Identificazione dei requisiti

- Le funzioni
  - che operazioni deve compiere il sistema?
- I dati
  - che dati sono manipolati? che entità applicative rappresentano?
- Il controllo
  - in che ordine sono eseguite le operazioni?
- Il tempo (per sistemi real-time)
  - quando? quali vincoli temporali?
- I diversi punti di vista dei diversi tipi di utenza
- I diversi punti di vista di diverse componenti del sistema

Identificazione: brainstorming



Servizi offerti ai viewpoints

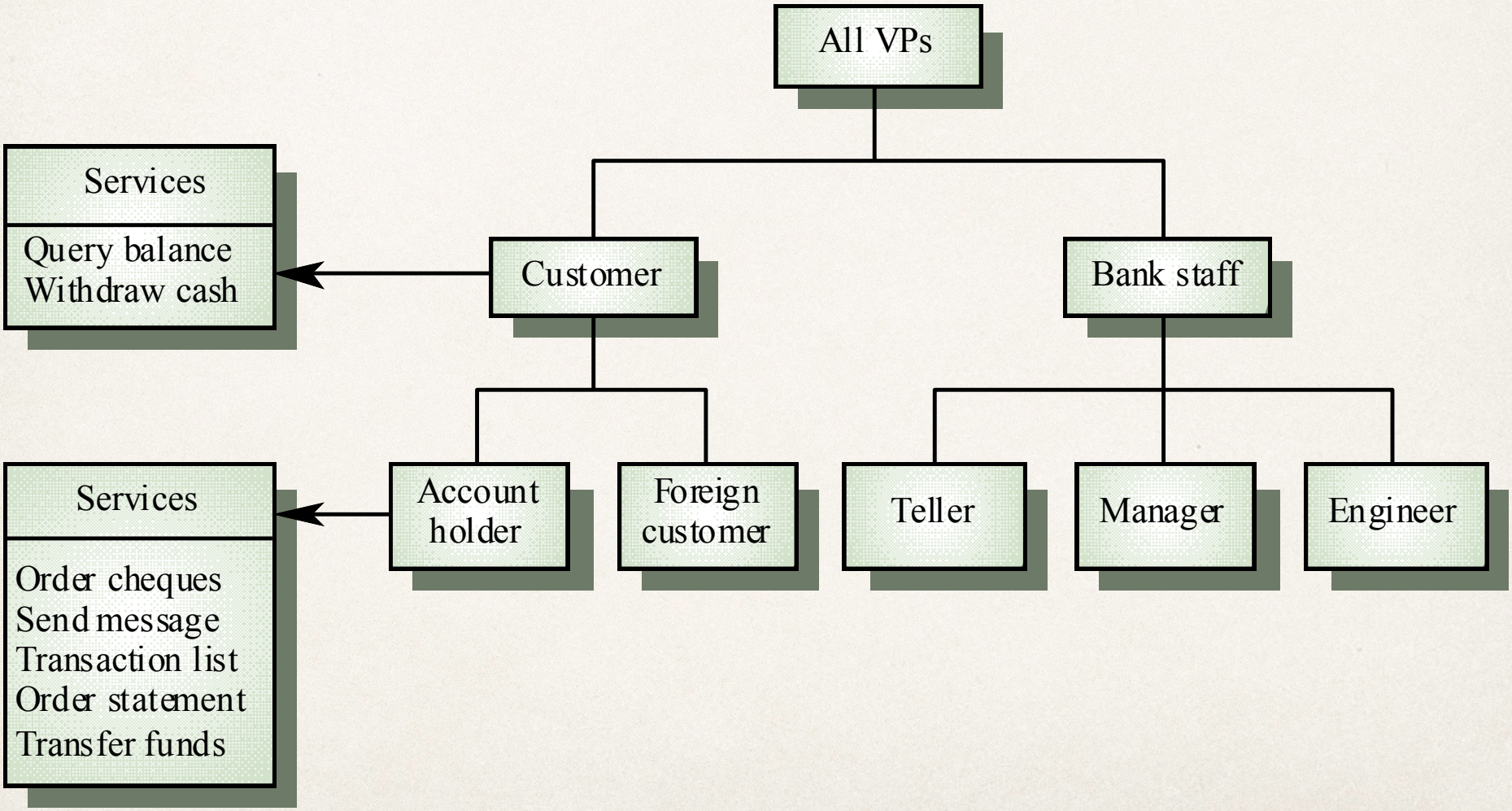


Dati in ingresso/uscita per ogni singolo viewpoint

ACCOUNT HOLDER

Control input	Data input
Start transaction Cancel transaction End transaction Select service	Card details PIN Amount required Message

Gerarchia dei viewpoints

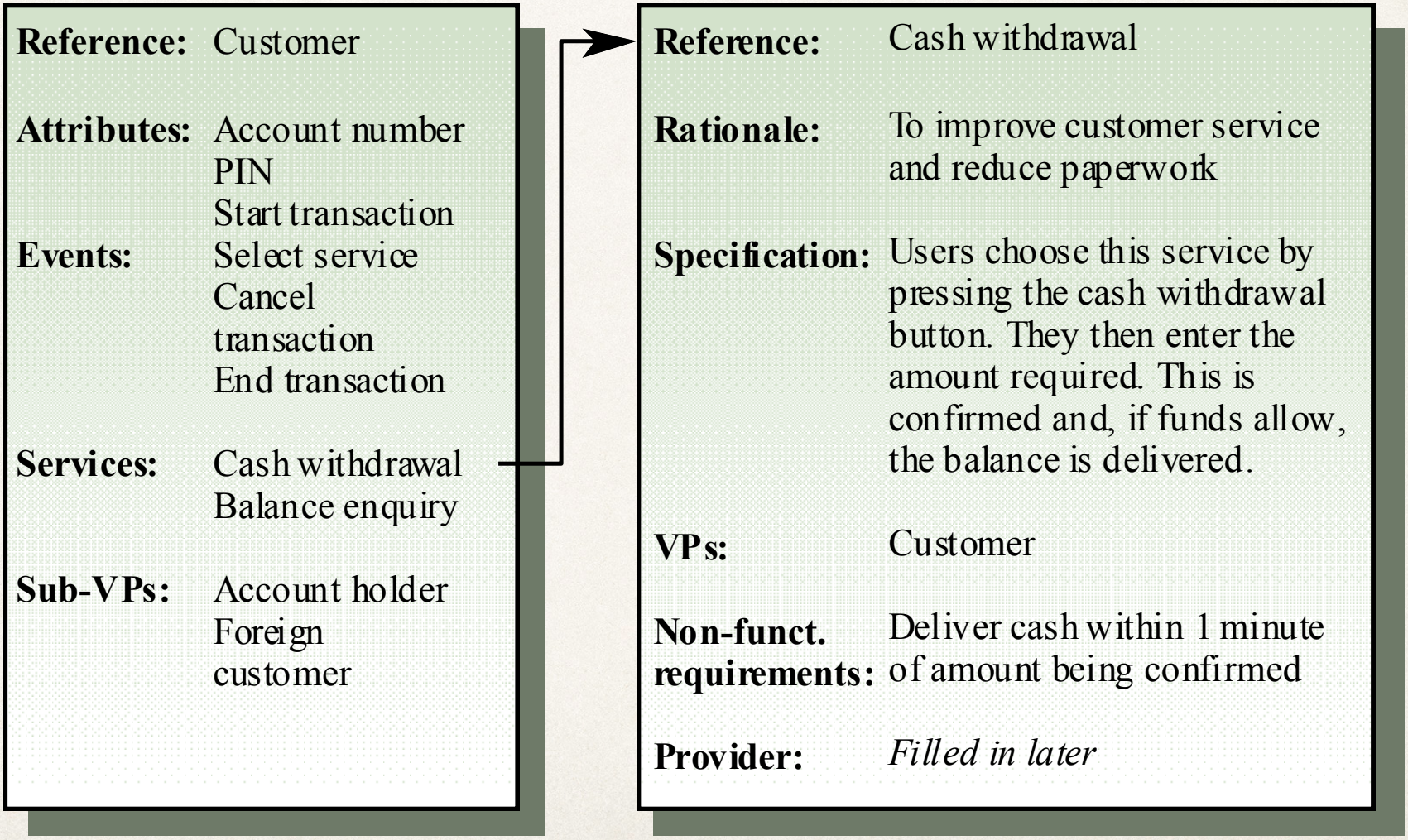


Moduli per la documentazione

Viewpoint template	
Reference:	The viewpoint name.
Attributes:	Attributes providing viewpoint information.
Events:	A reference to a set of event scenarios describing how the system reacts to viewpoint events.
Services	A reference to a set of service descriptions.
Sub-VPs:	The names of sub-viewpoints.

Service template	
Reference:	The service name.
Rationale:	Reason why the service is provided.
Specification:	Reference to a list of service specifications. These may be expressed in different notations.
Viewpoints:	List of viewpoint names receiving the service.
Non-functional requirements:	Reference to a set of non-functional requirements which constrain the service.
Provider:	Reference to a list of system objects which provide the service.

Esempio di moduli preparati



Azioni del viewpoint “cliente”

- Inserimento carta
  - ✓ input: richiesta carta
    - sorgente:terminale
  - ✓ output: carta
    - dest: terminale
- scrittura PIN
  - ✓ input: richiesta PIN
    - sorgente: terminale
  - ✓ output: PIN
    - dest: terminale
- raccolta contante
  - ✓ input: richiesta di raccolta contante
    - sorgente: terminale
  - ✓ output: contante
    - dest: cliente
  - ✓ output: conferma raccolta
    - dest: terminale

Azioni del viewpoint “terminale”

- convalida carta
  - ✓ input: carta
    - sorgente: cliente
  - ✓ output: carta
    - dest: cliente
  - ✓ output: msg errore
    - dest: cliente
  - ✓ output: richiesta PIN
    - dest: cliente
  - ✓ output: richiesta c/c
    - dest: database
- convalida PIN
  - ✓ input: PIN
    - sorgente: cliente
  - ✓ input: info su c/c
    - sorgente: database
  - ✓ output: carta
    - dest: cliente
  - ✓ output: msg errore
    - dest: terminale
  - ✓ output: richiesta servizio
    - dest: terminale

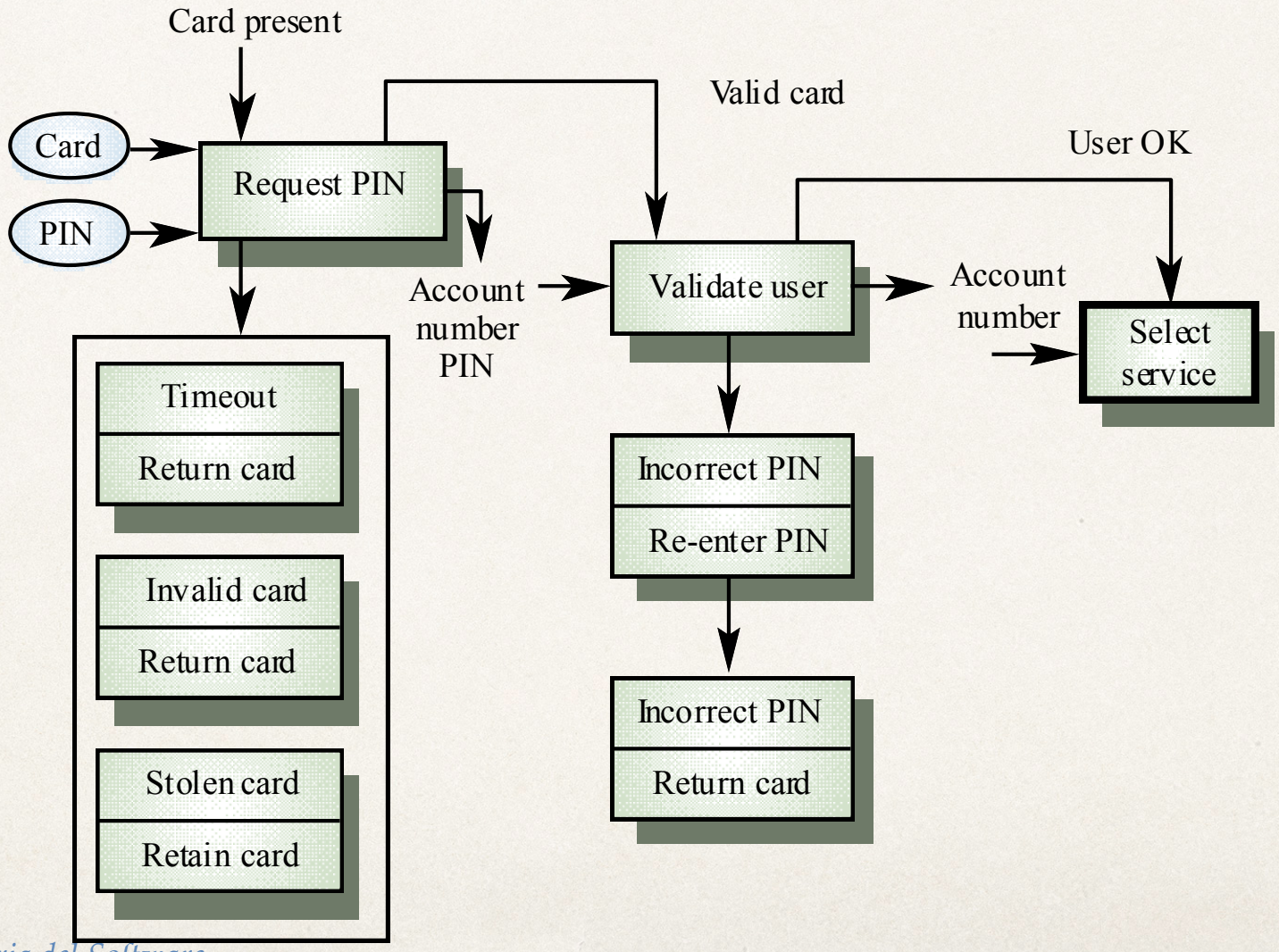
Azioni del viewpoint “terminale”

- verifica stato c/c
  - ✓ input: richiesta contante
    - sorgente: cliente
  - ✓ input: info c/c
    - sorgente: terminale
  - ✓ output: msg errore
    - destinazione: terminale
  - ✓ output: msg di conteggio contante
    - destinazione: terminale
  - ✓ output: contanti
    - destinazione: terminale

Analisi dei dati e controllo: la notazione

- ellissi: dati offerti o forniti da un viewpoint
- box: rappresenta un evento
- l'informazione di controllo entra ed esce dall'alto di un box
- i dati entrano da sinistra ed escono da destra
- le eccezioni sono poste sotto al box corrispondente

Analisi dei dati e controllo



Analisi dei dati e controllo: descrizione delle eccezioni

Nel caso in esempio

- Timeout
  L'utente non inserisce il PIN entro il tempo limite
- Carta non valida
  La carta bancaria non è riconosciuta e viene espulsa
- Carta rubata
  La carta è riconosciuta tra quelle dichiarata rubate/perse e viene trattenuta dalla macchina

Specifiche date in linguaggio naturale strutturato

- 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
  - indicazione di altre entità richieste
  - pre e post condizioni
  - effetti collaterali (se ce ne sono)
- N.B.: è come la specifica di uno use case

Esempio

Definition: 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 dispayed 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

Definizione dei requisiti con PDL

- I requisiti possono essere definiti in modo operazionale usando un linguaggio simile ad un linguaggio di programmazione ma più espressivo
- Appropriato in due casi
  - quando un'operazione è specificata come sequenza di azioni e l'ordine è importante
  - quando si devono specificare interfacce hw e sw
- Svantaggi
  - può non essere abbastanza espressivo da definire i concetti del dominio in modo adeguato
  - si può fare generare confusione tra specifica e progettazione

Esempio

- specifica di uno sportello bancomat

```
-ATM/RS/CONT/1  Control specification for an ATM
procedure ATM is
      PIN: Pin_no ;
      Acc_no: Account_number ;
      Balance: Amount ;
      Service: Available_services ;
      Valid_card, Valid_PIN: Boolean ;
begin
      loop
            Get_card ( Acc_no, PIN, Valid_card ) ;
            if Valid_card then
                  Validate_PIN ( PIN, Valid_PIN ) ;
                  if Valid_PIN then
                        Get_account (Acc_no, Balance);
                        Get_service(Service);
                        while a service is selected loop
                              Deliver_selected_service ;
                              Get_service(Service);
                        end loop ;
                        Return_card ;
                  end if ;
            end if ;
      end loop ;
end ATM ;
```