

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Definizione e specifica dei requisiti

- Tecniche per analizzare, definire e specificare i requisiti dei sistemi software

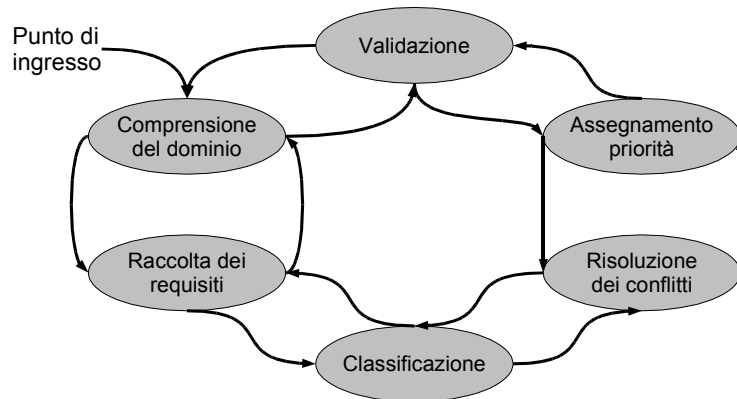
Analisi dei requisiti

- Coinvolge lo staff tecnico che collabora con il cliente per individuare il dominio applicativo, i servizi che il sistema deve fornire e i vincoli operativi
- Può coinvolgere utenti finali, manager, ingegneri coinvolti nella manutenzione, esperti del dominio...
 - In breve, queste sono dette le *controparti*

Problemi dell'analisi dei requisiti

- Le controparti non sanno esattamente cosa vogliono
- Le controparti esprimono i requisiti nella propria terminologia
- Diversi contraenti possono avere requisiti contrastanti
- Fattori politici e organizzativi possono influenzare i requisiti del sistema
 - I requisiti sono inevitabilmente soggetti al cambiamento. Nuovi contraenti possono emergere

Il processo di analisi dei requisiti



Attività del processo / 1

- **Comprensione del dominio**
 - Gli analisti devono sviluppare una propria comprensione del dominio applicativo. Se il sistema è un supermercato, gli analisti devono diventare "esperti" di supermercati
- **Raccolta dei requisiti**
 - Interagire con i contraenti per identificare i loro requisiti
- **Classificazione**
 - Prendere la collezione non strutturata dei requisiti ed organizzarli in cluster coerenti

Attività del processo / 2

- **Risoluzione dei conflitti**
 - Quanto molteplici contraenti sono coinvolti, possono sorgere conflitti nei requisiti. E' quindi necessario individuare e risolvere tali conflitti
- **Attribuzione delle priorità**
 - In ogni insieme di requisiti, alcuni saranno più importanti di altri. E' quindi necessario individuare le classi di priorità
- **Validazione dei requisiti**
 - Controllare i requisiti per capire se sono completi, consistenti e in accordo con ciò che i contraenti realmente vogliono dal sistema

Viewpoint-Oriented analysis

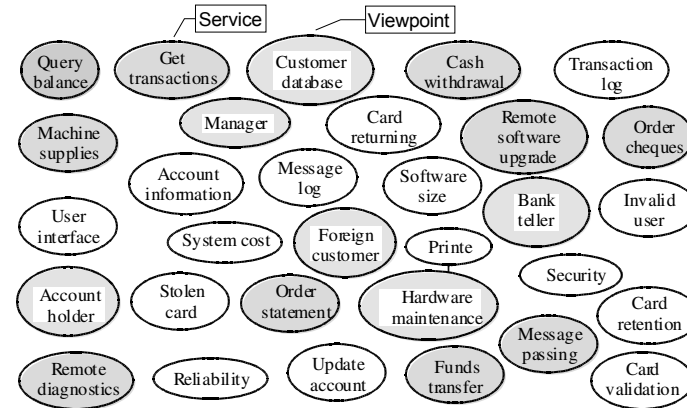
- I contraenti rappresentano diversi modi di guardare al problema
- Questa analisi "multi-prospettiva" è importante, dato che raramente c'è un unico modo "corretto" per analizzare i requisiti del sistema
 - Non c'è una singola vista corretta del sistema, ma ci sono tanti modi ugualmente corretti di guardare ad esso

VORD

(Viewpoint Oriented Requirements Definition)

- Identificazione dei viewpoint
 - Individuare i viewpoint che ricevono servizi dal sistema, e identificare i servizi forniti a ciascun viewpoint
- Strutturazione dei viewpoint
 - Rappresentare i viewpoint in una gerarchia. I servizi comuni sono forniti ai viewpoint che stanno nei livelli più alti della gerarchia
- Documentazione dei viewpoint
 - Raffinare la descrizione dei viewpoint e servizi identificati
- Mapping viewpoint-sistema
 - Definire gli oggetti che rappresentano i viewpoint

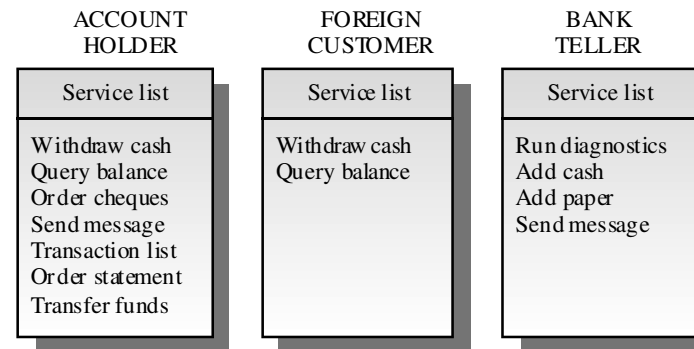
Identificazione dei viewpoint



Form standard del processo VORD

Viewpoint template		Service template	
Reference:	The viewpoint name.	Reference:	The service name.
Attributes:	Attributes providing viewpoint information.	Rationale:	Reason why the service is provided.
Events:	A reference to a set of event scenarios describing how the system reacts to viewpoint events.	Specification:	Reference to a list of service specifications. These may be expressed in different notations.
Services:	A reference to a set of service descriptions.	Viewpoints:	List of viewpoint names receiving the service.
Sub-VPs:	The names of sub-viewpoints.	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.

Associare viewpoint a servizi

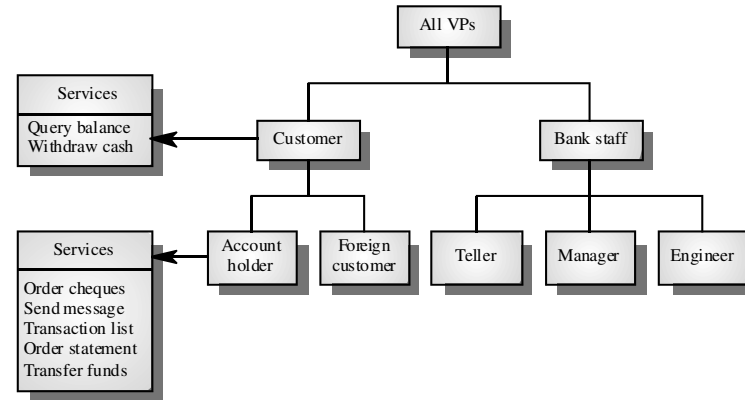


Dati in ingresso/uscita a ciascun viewpoint

ACCOUNT HOLDER

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

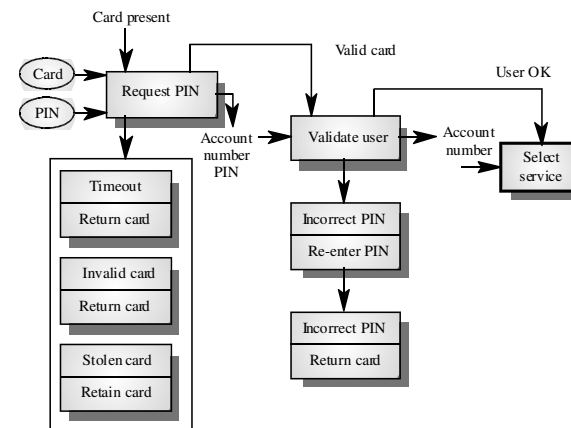
Gerarchia dei viewpoint



Un esempio di template

Reference: Customer	Reference: Cash withdrawal
Attributes: Account number PIN Start transaction Select service Cancel transaction End transaction	Rationale: To improve customer service and reduce paperwork
Events: Cash withdrawal Balance enquiry	Specification: Users choose this service by pressing the cash withdrawal button. They then enter the amount required. This is confirmed and, if funds allow, the balance is delivered.
Services: Account holder Foreign customer	VPs: Customer
Sub-VPs:	Non-funct. requirements: Deliver cash within 1 minute of amount being confirmed
	Provider: Filled in later

Dati e controlli



Definizione e Specifica dei requisiti

- Ricordiamo:
 - 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 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 (ad esempio verso gli utenti) SENZA definirne un modello computazionale
- Include requisiti funzionali e non funzionali
 - I requisiti funzionali descrivono le funzionalità che il sistema deve fornire
 - I requisiti non funzionali sono vincoli sui servizi e le funzionalità fornite, o sul processo di sviluppo o sull'ambiente esterno; i requisiti non funzionali includono anche le proprietà che il sistema deve o non deve avere

Come si scrive la definizione dei requisiti

- Di solito, in linguaggio naturale utilizzando anche tabelle e diagrammi. Quindi in modo informale
- Questo è universalmente comprensibile, ma ci sono almeno tre problemi
 - Mancanza di chiarezza. E' difficile descrivere con precisione senza rendere il documento illeggibile
 - Confusione dei requisiti. I requisiti funzionali e non funzionali tendono ad essere mescolati
 - Amalgama dei requisiti: Requisiti diversi potrebbero essere descritti insieme (ad esempio, nella stessa frase)

Esempio: ambiente di supporto alla programmazione in linguaggio ADA

4.A.5 The database shall support the generation and control of configuration objects; that is, objects which are themselves groupings of other objects in the database. The configuration control facilities shall allow access to the objects in a version group by the use of an incomplete name.

Problema

- Mescola informazioni astratte e dettagliate
 - Descrive che dovrebbe esserci una facility per il controllo della configurazione
 - Ma include anche il dettaglio che queste facilities devono consentire l'accesso agli oggetti appartenenti ad un gruppo di revisione (*version* group) utilizzando un nome parziale

Griglia di un editor visuale

2.6 Grid facilities To assist in the positioning of entities on a diagram, the user may turn on a grid in either centimetres or inches, via an option on the control panel. Initially, the grid is off. The grid may be turned on and off at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines.

Problemi

- Requisiti funzionali e non funzionali sono mescolati
 - Funzionale (con giustificazione): Deve esserci una griglia
 - Non funzionale: L'unità di misura della griglia
 - Interfaccia, non-funzionale: Come si comporta la griglia
- E' facile criticare i requisiti mal scritti, ma è difficile scriverli correttamente
- L'uso di un formato standard con campi predefiniti da riempire implica solitamente una migliore strutturazione

Esempio

2.6 Grid facilities

2.6.1 The editor shall provide a grid facility where a matrix of horizontal and vertical lines provide a background to the editor window. This grid shall be a passive grid where the alignment of entities is the user's responsibility. *Rationale:* A grid helps the user to create a tidy diagram with well-spaced entities. Although an active grid, where entities 'snap-to' grid lines can be useful, the positioning is imprecise. The user is the best person to decide where entities should be positioned.

2.6.2 When used in 'reduce-to-fit' mode (see 2.1), the number of units separating grid lines must be increased.

Rationale: If line spacing is not increased, the background will be very cluttered with grid lines.

Specification: ECLIPSE/WS/Tools/DE/FS Section 5.6

Giustificazione dei requisiti (*rationale*)

- E' importante fornire una giustificazione ai requisiti (perché un tal requisito è presente)
- Aiuta gli sviluppatori a comprendere il dominio applicativo e perché il requisito viene dato in quel modo particolare
- Ciò è particolarmente utile nel caso in cui i requisiti debbano essere cambiati. Avere la giustificazione a disposizione riduce la possibilità che i cambiamenti abbiano effetti collaterali

Specificazione dei requisiti

- La specificazione dei requisiti aggiunge dettagli alla definizione dei requisiti; comunque deve essere consistente con essa
- Di solito anche la specificazione è scritta in linguaggio naturale, ma ciò è particolarmente problematico
 - Il linguaggio naturale assume che chi scrive e chi legge usino le stesse parole per esprimere gli stessi concetti
 - Una specificazione in linguaggio naturale è troppo flessibile nel senso che ammette diverse interpretazioni
 - E' difficile trovare requisiti collegati tra di loro

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

Alternative al linguaggio naturale

- Linguaggio naturale "strutturato"
 - Definire forme standard o template per esprimere le specifiche
- Design description languages
 - Una sorta di linguaggi di programmazione "maggiormente astratti". Definiscono i requisiti fornendo una visione operativa del sistema
- Requirements specification languages
- Notazioni grafiche
- Specifiche matematiche

Riepilogo notazioni alternative al linguaggio naturale

Notation	Description
Structured natural language	This approach depends on defining standard forms or templates to express the requirements specification.
Design description languages	This approach uses a language like a programming language but with more abstract features to specify the requirements by defining an operational model of the system. This approach is not now widely used although it can be useful for interface specifications.
Graphical notations	A graphical language, supplemented by text annotations is used to define the functional requirements for the system. An early example of such a graphical language was SADT. Now, use-case descriptions and sequence diagrams are commonly used.
Mathematical specifications	These are notations based on mathematical concepts such as finite-state machines or sets. These unambiguous specifications reduce the arguments between customer and contractor about system functionality. However, most customers don't understand formal specifications and are reluctant to accept it as a system contract.

Tracciabilità dei requisiti

- Significa che 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
 - Esempio: trovare tutti i requisiti che contengono la stessa parola chiave
- Di solito la tracciabilità si ottiene numerando i requisiti, e utilizzando tali numeri per inserire riferimenti incrociati o creare indici

Specificata data in linguaggio naturale strutturato

- Utilizzare una forma molto limitata di un linguaggio naturale per esprimere i requisiti
 - Evitare costrutti ambigui, perifrasi, arricchimenti linguistici
 - Usare una esposizione asciutta e chiara
- Ciò aggiunge uniformità alla specifica
- Di solito un linguaggio strutturato viene usato naturalmente per compilare form

Esempio di form

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
<i>Definition: ECLIPSE/Workstation/Tools/DE/RD/3.5.1</i>	

Program Description Language (PDL)

- I requisiti possono essere definiti *operazionalmente* utilizzando una specie di linguaggio di programmazione, solo molto meno vincolato
- Adatto in due situazioni
 - Quando una operazione deve essere specificata come una sequenza di azioni, e l'ordine è importante
 - Quando è necessario specificare le interfacce tra hardware e software

Svantaggi del PDL

- I PDL possono essere insufficienti per descrivere informazioni ad alto livello (esempio, requisiti non funzionali, vincoli, informazioni sul dominio applicativo)
- La specifica dei requisiti tenderà ad essere considerata un documento di design, piuttosto che una specifica dei requisiti

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 ;
```

Specifica delle interfacce

- Molti sistemi software operano in un ambiente che include altri sistemi. Possono doversi interfacciare con questi sistemi in diversi modi
- Tre tipi di interfacce possono dover essere definiti in un documento di specifica dei requisiti
 - Interfacce procedurali: servizi offerti da sottosistemi
 - Interfacce dei dati: strutture dati che vengono trasmesse
 - Interfacce di rappresentazione: specifici pattern utilizzati per descrivere dati

Interfaccia procedurale

```
package Print_server is  
  procedure Initialise (P: PRINTER) ;  
  procedure Print (P: PRINTER ; F: PRINT_FILE ) ;  
  procedure Display_print_queue (P: PRINTER ) ;  
  procedure Cancel_print_job (P: PRINTER; N: PRINT_ID) ;  
  procedure Switch_printer (P1, P2: PRINTER; N: PRINT_ID) ;  
end Print_server ;
```

Interfaccia dei dati

```
type MESSAGE is record  
  Sender : SYSTEM_ID;  
  Receiver : SYSTEM_ID;  
  Dispatch_time : DATE;  
  Length: MESSAGE_LENGTH ;  
  Terminator: CHARACTER ;  
  Message : TEXT;  
end record;  
type SYSTEM_ID is range 20_000..30_000 ;  
type YEAR_TYPE is range 1980..2080 ;  
type DATE is record  
  Seconds: NATURAL ;  
  Year: YEAR_TYPE ;  
end record ;  
type MESSAGE_LENGTH is range 0..10_000 ;  
type TEXT is array (MESSAGE_LENGTH) of CHARACTER ;
```

Interfaccia di rappresentazione

```
for SYSTEM_ID'SIZE use 2*BYTE ;  
for YEAR_TYPE'SIZE use 2*BYTE ;  
for MESSAGE_LENGTH'SIZE use 2*BYTE ;  
type STATE is (Halted, Waiting, Ready, Running);  
for STATE use (Halted => 1, Waiting => 4, Ready => 16,  
              Running => 256);
```

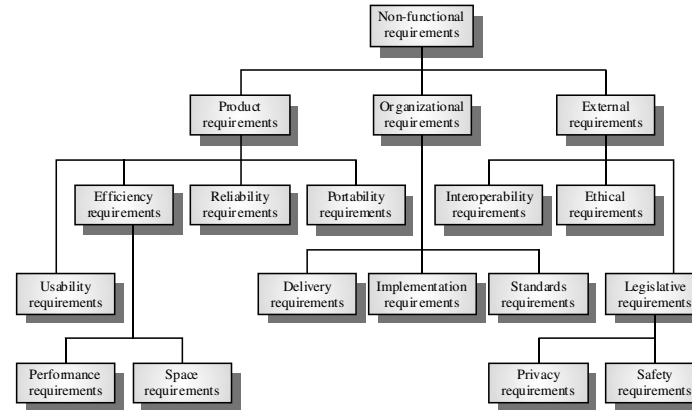
Requisiti non funzionali

- Definiscono proprietà del sistema e vincoli (ad es. affidabilità, tempo di risposta, capacità di immagazzinamento dati).
- I requisiti non funzionali possono essere più critici dei requisiti funzionali.
 - Se certi requisiti non funzionali non fossero soddisfatti, il sistema potrebbe risultare inutilizzabile

Classificazione dei requisiti non funzionali

- **Requisiti del prodotto**
 - Specificano che il prodotto consegnato deve comportarsi in determinati modi, ad es. velocità di esecuzione, affidabilità, innocuità...
- **Requisiti organizzativi**
 - Sono conseguenze di politiche organizzative e procedure interne, ad es. standard cui ci si deve conformare, linguaggi di programmazione o metodi da adottare, tempi di consegna del prodotto e della documentazione...
- **Requisiti esterni**
 - Es. interoperabilità, requisiti legislativi...

Schema dei requisiti non funzionali



Esempi di requisiti non funzionali

- **Product requirement**
 - 4.C.8 *It shall be possible for all necessary communication between the Ada Programming Support Environment (APSE) and the user to be expressed in the standard Ada character set.*
- **Organisational requirement**
 - 9.3.2 *The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.*
- **External requirement**
 - 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 tale da essere oggettivamente verificabili
- Evitare termini come “minimizzare l'errore”
 - *"The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised."*
- Meglio quantificare
 - *"Experienced controllers should be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users should not exceed two per day."*

Misure dei 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