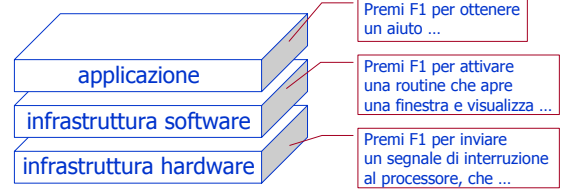


Capitolo 6 – Le applicazioni

Introduzione ai sistemi informatici

L'organizzazione "per livelli" dei sistemi informatici



... sono "punti di vista" diversi, e complementari, sullo stesso sistema...

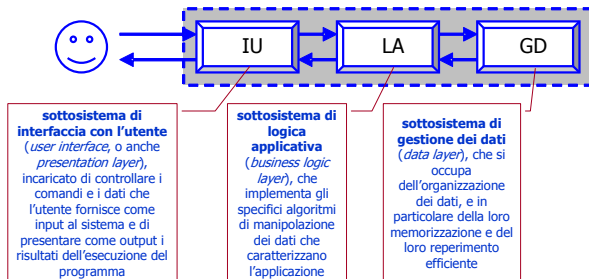
Un'applicazione ben costruita (e funzionante...!) può essere usata prescindendo in larga misura dalle problematiche infrastrutturali dei livelli sottostanti

26/10/02

Introduzione ai sistemi informatici

2

L'organizzazione funzionale di un'applicazione informatica

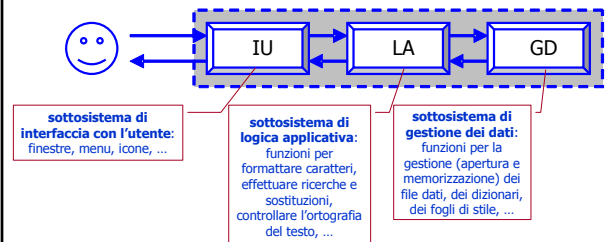


26/10/02

Introduzione ai sistemi informatici

3

Organizzazione funzionale delle applicazioni: l'esempio dei sistemi di elaborazione di testi



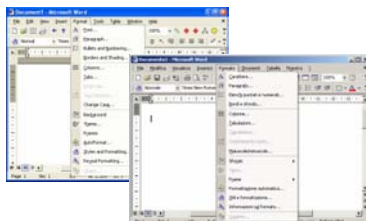
26/10/02

Introduzione ai sistemi informatici

4

Organizzazione funzionale delle applicazioni: indipendenza funzionale

I tre sottosistemi sono funzionalmente indipendenti l'uno dall'altro...
... si può, per esempio, cambiare IU senza modificare LA e GD



... come nel caso della localizzazione delle applicazioni...

26/10/02

Introduzione ai sistemi informatici

5

Sviluppare è più che programmare

Il processo di sviluppo delle applicazioni è tipicamente scandito nelle seguenti attività:

- **analisi e specifica dei requisiti** → studio del dominio del problema e dei requisiti dei potenziali utenti
la specifica dei requisiti varia in base alla tipologia di prodotto:
* per progetti interni a organizzazioni, il progetto nasce da esigenze ben definite, che possono richiedere tempo per la formalizzazione, ma che derivano da uno specifico committente
* per prodotti commerciali, da lanciare sul mercato, l'utente inizialmente non esiste, ma sta allo sviluppatore cercare di identificare, con le opportune tecniche, le esigenze dei possibili utenti, in modo da rendere il prodotto interessante al maggior numero di potenziali acquirenti
il risultato è un documento che descrive i requisiti identificati
- **progettazione** → studio della soluzione informatica al problema da risolvere, per come identificato dall'analisi dei requisiti
il risultato è un documento che specifica l'architettura software da sviluppare, definita mediante un linguaggio di specifiche formale o semiformale
- **sviluppo e test dell'applicazione software** → l'implementazione delle componenti software in base alle specifiche di progetto, impiegando il linguaggio di programmazione scelto, e nella verifica che il codice sviluppato rispetti i requisiti stabiliti dalle specifiche
il risultato è la creazione dell'applicazione
- **rilascio (deployment)**
il risultato è la distribuzione dell'applicazione presso gli utenti
- **manutenzione**
il risultato è il mantenimento del corretto funzionamento dell'applicazione

26/10/02

Introduzione ai sistemi informatici

6

Il ciclo di vita di un'applicazione

Strutturabile secondo modelli diversi:

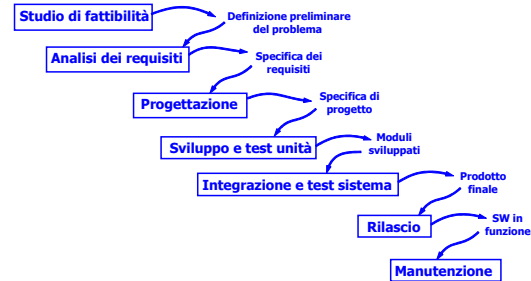
- **modello a cascata** → le attività vengono realizzate in sequenza lineare e ogni attività produce dei risultati utili allo svolgimento dell'attività successiva
- **modello a prototipi** → si comincia lo sviluppo direttamente con la realizzazione di prototipi del prodotto finale, con lo scopo di fornire informazioni per lo sviluppo definitivo
- **modello evolutivo** → si opera in modo incrementale, partendo dallo sviluppo delle componenti più critiche e arricchendo il prodotto man mano con nuove funzionalità, anche in base ai commenti ricevuti dagli utenti

26/10/02

Introduzione ai sistemi informatici

7

Il ciclo di vita di un'applicazione: il modello a cascata

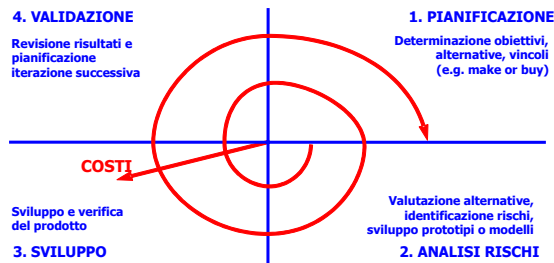


26/10/02

Introduzione ai sistemi informatici

8

Il ciclo di vita di un'applicazione: il modello evolutivo

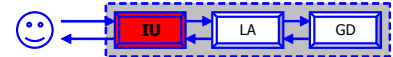


26/10/02

Introduzione ai sistemi informatici

9

Il sottosistema IU – interfaccia utente



Ha il compito di gestire l'interazione del programma con l'utente:

- consente all'utente di specificare gli input al programma
- restituisce all'utente gli output del programma
- e inoltre:
- fornisce informazioni sullo stato di esecuzione del programma

A causa dell'indipendenza funzionale tra sottosistemi, le stesse caratteristiche per LA e GD possono essere ottenute con IU diverse...

26/10/02

Introduzione ai sistemi informatici

10

Il sottosistema IU – Un esempio /1

LA: contare il numero n di volte che un carattere x è presente in una stringa y ; dunque, input= x , y ; output= n (p.es. $x='e'$, $y='E'$ l'esempio di un programma' → $n=2$)

IU (opzione 1): interfaccia a linea di comando (*command line*), con modalità batch

```
>contal
Conta il numero di occorrenze di un carattere all'interno di una stringa.
Se viene specificato il terzo argomento, -n, il conteggio viene effettuato
senza distinguere tra maiuscole e minuscole.

Uso: contal -c<carattere> -s"<stringa>" [-n]

>contal -c'e' -s'E' l'esempio di un programma"
Numero di occorrenze: 2
>_
```

Questo programma opera secondo la logica:

1. l'utente specifica tutti i parametri di input e mette in esecuzione il programma
2. il programma viene eseguito
3. il risultato prodotto nell'esecuzione viene presentato all'utente come output del programma
4. l'esecuzione termina

26/10/02

Introduzione ai sistemi informatici

11

Il sottosistema IU – Un esempio /2

IU (opzione 2): interfaccia a linea di comando (*command line*), con modalità interattiva modale

```
>contal2
Conta il numero di occorrenze di un carattere all'interno di una stringa.

Carattere: a
Stringa: Ancora una prova
Distinguere tra minuscole e minuscole <s/n>?_

Numero di occorrenze: 4
>_
```

Questo programma opera secondo la logica:

1. l'utente mette in esecuzione il programma
2. il programma attiva una sessione domande-risposte per ottenere dall'utente i parametri di input
3. viene eseguita la parte del programma che opera sui parametri di input per ottenere il risultato
4. il risultato prodotto nell'esecuzione viene presentato all'utente come output del programma
5. l'esecuzione termina

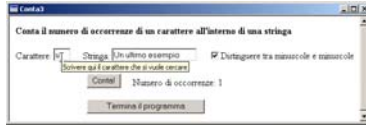
26/10/02

Introduzione ai sistemi informatici

12

Il sottosistema IU – Un esempio /3

IU (opzione 3): interfaccia **grafica** (*graphical user interface*),
con **modalità interattiva non modale**



Questo programma opera secondo la logica:

1. l'utente mette in esecuzione il programma
2. il programma attiva un'interfaccia utente interattiva non modale e si mette in condizione "di ascolto"
3. fino a che l'utente non seleziona il pulsante per terminare l'esecuzione del programma:
 - a) l'utente immette i parametri di input
 - b) l'utente seleziona il pulsante che mette in esecuzione la parte del programma che opera sui parametri di input per ottenere il risultato
 - c) il risultato prodotto viene presentato all'utente
4. l'esecuzione termina

26/10/02

Introduzione ai sistemi informatici

13

Graphical user interface (GUI): personalizzazione

Forniscono una molteplicità di strumenti di input...

(p.es. per copiare nell'archivio appunti (*clipboard*) un testo selezionato si può:

- (1) con la tastiera premere Ctrl-C
- (2) con la tastiera aprire il menu Modifica con il corrispondente tasto scorciatoia e quindi muovere il cursore con i tasti freccia fino a evidenziare la voce Copia premendo quindi Invio
- (3) con il mouse aprire il menu Modifica e quindi selezionare la voce Copia
- (4) con il mouse aprire il menu contestuale (cursore sopra la selezione e click con il tasto di destra) da cui scegliere la voce Copia
- (5) ...

... e consentono quindi a ogni utente di adottare
un proprio **stile di uso** dei programmi e, spesso,
di **personalizzarne l'interfaccia**

26/10/02

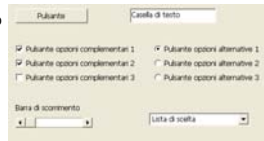
Introduzione ai sistemi informatici

14

Graphical user interface (GUI): standardizzazione

L'uniformità delle IU consente agli utenti di focalizzare l'attenzione sulle funzionalità del programma e non sull'interfaccia utente (sul **cosa** fare e non sul **come** farlo); sono stati standardizzati:

- > l'uso di sequenze di tasti di comando, per esempio F1 per richiamare un aiuto sul programma in esecuzione o Ctrl-C per copiare nell'archivio appunti l'entità selezionata
- > la presenza e le modalità di uso dell'archivio appunti, che funziona in modo omogeneo in diverse applicazioni e consente di trasferire dati tra applicazioni
- > la presenza e le modalità di uso di funzioni di utilità di impiego generale, come Apri..., Salva con nome..., o Stampa...
- > la presentazione grafica e le modalità di uso degli oggetti grafici nelle finestre:

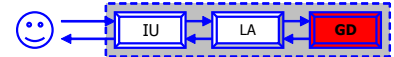


26/10/02

Introduzione ai sistemi informatici

15

Il sottosistema GD – gestione dei dati



Ha il compito di organizzare i dati gestiti dall'applicazione, in particolare relativamente alla loro memorizzazione e al loro reperimento efficiente

A causa dell'indipendenza funzionale tra sottosistemi, le stesse caratteristiche per IU e LA possono essere ottenute con GD diversi...

Per esempio, una tipica applicazione di elaborazione di testi mette a disposizione diversi formati di file per memorizzare i testi

Il sottosistema GD diventa critico quando utenti che usano applicazioni diverse vogliono **scambiarsi dei dati**

26/10/02

Introduzione ai sistemi informatici

16

Il sottosistema GD – Scambio di dati

Scenari (l'utente B deve operare su un file inizialmente prodotto dall'utente A):

1. i programmi di A e B adottano formati di file non compatibili e non offrono strumenti per rendere i dati accessibili all'esterno: **B deve riscrivere il testo**
2. i programmi di A e B adottano formati di file non compatibili ma dispongono di filtri di esportazione e/o di importazione: **A può esportare il testo nel formato del programma di B oppure B può importare il testo dal formato del programma di A**
3. i programmi di A e B adottano uno stesso formato di file e quindi si comportano come se disponessero di uno stesso sottosistema GD: **non ci sono problemi di scambio dei dati**

26/10/02

Introduzione ai sistemi informatici

17

Il sottosistema GD – Il ruolo dei DBMS

Una tecnologia consolidata per la gestione dei dati persistenti (cioè che devono essere mantenuti in modo permanente, e non solo durante l'esecuzione del programma): **sistemi di gestione di basi di dati** (DBMS, *data base management system*)

In particolare i DBMS basati sul modello relazionale forniscono un'interfaccia comune per la definizione della struttura del DB, l'amministrazione del DB stesso e l'accesso ai dati, ottenuta con l'impiego del linguaggio SQL

Grazie a SQL:

- > i dati mantenuti in un DBMS possono essere manipolati da programmi molteplici (*accessibilità*)
- > i dati possono essere trasferiti tra DBMS diversi e rimanere identicamente accessibili (*indipendenza dal DBMS*)

26/10/02

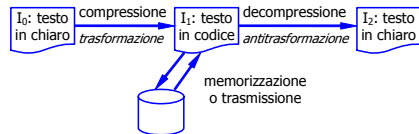
Introduzione ai sistemi informatici

18

I servizi di compressione dei dati

Nel caso di:

- memorizzazione su un supporto di capacità scarsa
 - trasmissione su un canale di capacità scarsa
- si impiegano tipicamente **servizi di compressione dei dati**



26/10/02

Introduzione ai sistemi informatici

19

Applicazioni come sistemi

Le applicazioni tradizionali vengono realizzate come moduli software che si basano direttamente sui servizi offerti dal sistema operativo

Le applicazioni recenti sono invece spesso basate sull'assemblaggio di componenti messi a disposizione da uno strato di *middleware*



26/10/02

Introduzione ai sistemi informatici

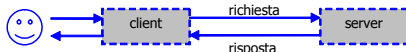
20

Applicazioni come sistemi distribuiti

I sottosistemi di cui un'applicazione è costituita possono risiedere (ed essere in esecuzione) su calcolatori diversi connessi in rete: si parla in tal caso di **applicazioni distribuite**

L'architettura più diffusa per la realizzazione di applicazioni distribuite è chiamata **client-server**:

- il programma client, con cui opera l'utente, include il solo sottosistema IU; grazie a esso l'utente produce una **richiesta** da inviare al server e ottiene da esso la relativa **risposta**
- il programma server, in esecuzione su un calcolatore raggiungibile sulla rete, include i sottosistemi LA e GD; ricevendo una **richiesta** dalla rete, esso la elabora e produce una **risposta**, che reinvia sulla rete



26/10/02

Introduzione ai sistemi informatici

21

Architettura client-server: un esempio

Un "sommatore distribuito":

- interagendo con il client l'utente specifica due numeri
- il client attiva la richiesta al server, passandogli i due numeri come parametri
- il server riceve la richiesta e attiva il suo sottoprogramma sommatore, passandogli i due parametri ricevuti
- il sottoprogramma sommatore produce il risultato e lo rende disponibile
- il server invia tale risultato come risposta al client
- il client riceve la risposta e ne visualizza il contenuto per l'utente



26/10/02

Introduzione ai sistemi informatici

22

Web come esempio di sistema distribuito

World Wide Web è realizzato come sistema distribuito client-server, in cui il client è un browser e il server è un server web:

- l'utente si connette a Internet e mette in esecuzione il browser; quindi scrive nella barra dell'indirizzo:
`http://www.curioso.it/CoseInteressanti/BelloQuesto.txt`
- e preme Invio
- il browser genera automaticamente un testo del tipo:
`GET /CoseInteressanti/BelloQuesto.txt HTTP/1.1`
`User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)`
`Accept: */*`
- e lo invia al server `www.curioso.it` come richiesta
- ricevendo la richiesta, il server web cerca il file richiesto (`/CoseInteressanti/BelloQuesto.txt`) e, trovandolo, produce un messaggio del tipo:
`HTTP/1.1 200 OK`
`Server: Microsoft-IIS/5.0`
`content-type: text/plain`
`content-length: 14`
`Saluti a tutti`
- che invia come risposta al browser da cui aveva ricevuto la richiesta
- il browser visualizza il contenuto del file, accodato all'intestazione della risposta

26/10/02

Introduzione ai sistemi informatici

23

HTTP come esempio di protocollo applicativo

Nella comunicazione tra client e server non è sufficiente che richiesta e risposta giungano correttamente al destinatario; occorre che i due programmi comunicanti siano in grado di comprendere le rispettive comunicazioni, cioè "parlino la stessa lingua": devono perciò condividere uno stesso **protocollo applicativo**

Nel caso di web il protocollo applicativo, che specifica il formato dell'intestazione sia della richiesta del browser sia della risposta del server web, è *HyperText Transfer Protocol* (HTTP)

Per esempio la risposta:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
content-type: text/plain
content-length: 14
Saluti a tutti
```

Intestazione HTTP, con informazioni sul server web che ha prodotto la risposta e sul file restituito

Linea bianca: separatore tra intestazione e corpo del messaggio

Corpo del messaggio: contenuto del file restituito

26/10/02

Introduzione ai sistemi informatici

24

Il problema dell'identificazione delle risorse

Per poter accedere ai servizi applicativi messi a disposizione dai server sulla rete e creare quindi un'applicazione distribuita, è necessario che ognuno di tali servizi risulti univocamente identificabile da parte dei client.

Nel caso di Internet si è standardizzato uno *schema uniforme di identificazione applicativa*, che consente di assegnare un metodo di accesso e un indirizzo, chiamati congiuntamente *Uniform Resource Locator (URL)*, a ogni risorsa presente sulla rete.

In generale, un URL ha la forma:

`protocollo://host:portaTCP/risorsa`

Per esempio:

`http://server1.isttec.liuc.it:80/dida/calendario.htm`

protocollo applicativo
adottato

indirizzo del server
web a cui è inviata la
richiesta

porta TCP su cui è
in esecuzione il
server web

nome completo
del file richiesto