

## La progettazione del software (design)

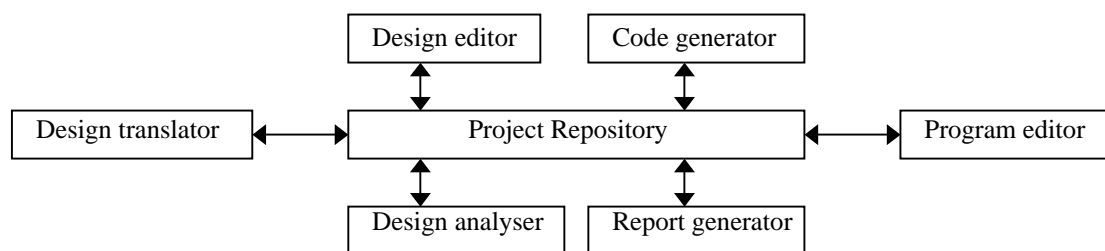
Nella fase di progettazione di un software si cercano di specificare tutte le strutture dati, gli algoritmi e i moduli del software che verranno utilizzati per implementare quest'ultimo, si cerca in altri termini di progettare l'architettura da utilizzare. E' possibile identificare cinque fasi del procedimento per la progettazione dell'architettura da utilizzare:

- 1) Le specifiche astratte, in cui si prende in esame il software e lo si scompone in sottomoduli, eseguendo poi una descrizione astratta, riprendendo i requisiti, tentando di specificarli, estrarli e descriverli in modo astratto, utilizzando Dataflow, modelli semantici, ecc.
- 2) La progettazione dell'interfaccia, in cui si focalizza in che modo interagiscono i diversi sottomoduli;
- 3) La progettazione dei componenti in cui è possibile osservare in che modo sono definiti i componenti dei sottomoduli;
- 4) La progettazione dei dati, in cui si definiscono i dati utilizzati;
- 5) La progettazione degli algoritmi, in cui si elencano gli algoritmi da utilizzare.

Durante la fase di progettazione è possibile definire più soluzioni, in modo da poter scegliere la soluzione ottimale. A questo proposito è bene ricordare che è possibile seguire diverse architetture:

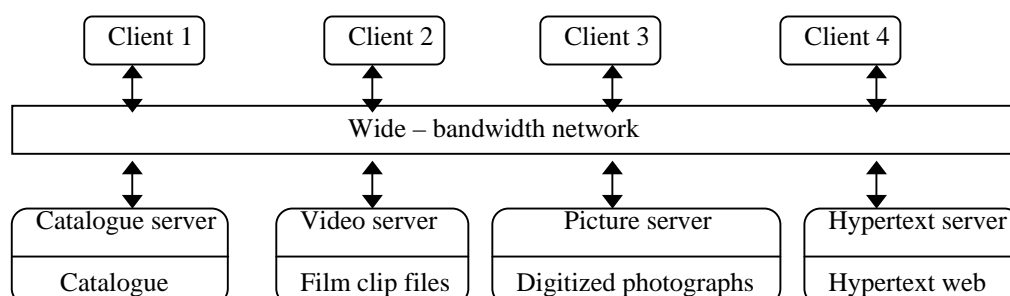
- 1) Il modello Repository, in cui i sottomoduli hanno in comune un database (= repository), ciascun sottomodulo gestisce i propri dati, scambiando poi informazioni, attraverso l'utilizzo di messaggi, con gli altri sottomoduli mediante il repository. Lo svantaggio di questa soluzione è rappresentato dalla modifica alle informazioni comuni, della quale ciascun sottomodulo dovrà esserne a conoscenza;

The architecture of an integrated CASE toolset



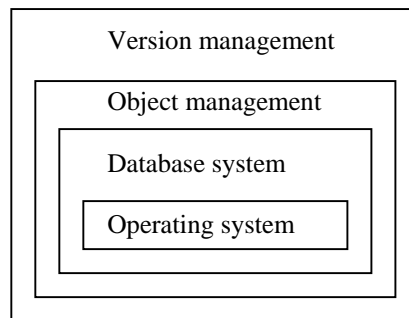
- 2) Il modello Client – Server, in cui ciascun modulo delega ad un server diversi compiti. Lo svantaggio di questa soluzione è che a volte il server non conosce il formato dei dati e ciò può comportare incertezze;

The architecture of a film and picture library system



3) Il modello Abstract Machine, basato sulla filosofia delle macchine virtuali;

Abstract machine model of a version management system.



4) L'architettura a controllore, in cui un controllore verifica se tutto procede bene. E' possibile seguire due soluzioni che implementano quest'architettura:

- a) l'utilizzo di un controllore che lavora in modo gerarchico;
- b) l'utilizzo di un controllore di tipo strutturato / centralizzato, attraverso cui tutti i sottomoduli dovranno passare per comunicare;

Nel secondo caso è possibile anche avere un controllore di tipo event – handler, il quale comunica a ciascun sottosistema il proprio compito attraverso la gestione di eventi. I sottosistemi risultano in entrambi i casi indipendenti, tuttavia, se il controllore rileva qualcosa di insolito, in questo caso potrebbe decidere di bloccare l'intero processo.

Nella progettazione bisogna osservare i requisiti in modo più dettagliato: in tal caso si possono utilizzare dei grafici per esprimere un flusso di dati, si possono utilizzare i linguaggi strutturati, per mezzo anche di pseudo - codice, o anche di linguaggio naturale.

### **La qualità della progettazione**

La qualità della progettazione è legata alle metriche della stessa fase. Con la coesione vengono distribuiti i ruoli di ogni modulo. Si cerca di effettuare una certa distribuzione logica tra i diversi compiti. Attraverso il coupling (o accoppiamento), è possibile individuare degli espedienti per rendere il programma indipendente ad esempio da valori fissi. Con l'adattabilità si pensa al futuro dei dati che compongono il software, offrendo così una struttura flessibile. E' possibile valutare l'adattabilità misurando la modularità di un sistema. Infine la progettazione deve essere comprensibile, cioè la documentazione, l'indentazione, i dataflow, ecc. devono essere comprensibili