

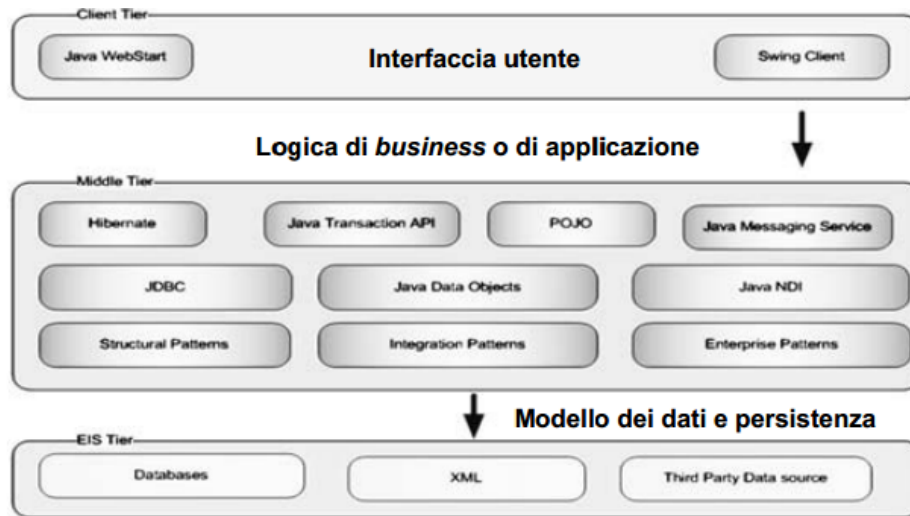
Lezione #19

Mercoledì, 20 Novembre 2013

Tullio Vardanega, 11:30-13:15

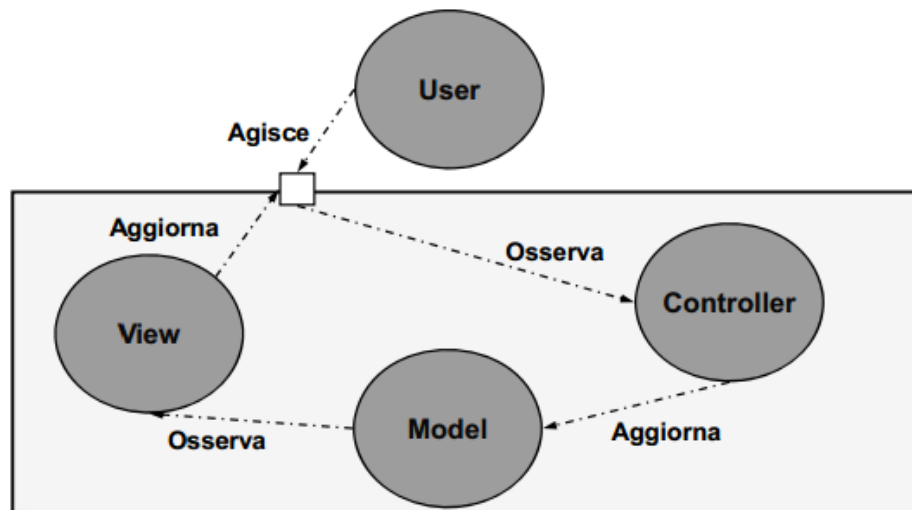
Luca De Franceschi

Facciamo alcuni esempi di architetture sw. Il progettista fissa quanti livelli servono e le interfacce che interagiscono tra questi livelli. Fissa una regola di dialogo sui livelli in modo da avere interazioni utili. Dentro ciascuna scatola le informazioni sono completamente incapsulate.



Tra l'interfaccia utente ed il modello dei dati c'è un supporto a specifiche procedure che forniscono la logica d'applicazione. Si vorrebbe scrivere una logica applicativa che non dipenda dalla base dati specifica, e questa cosa ce la fornisce il **framework**.

Architettura di relazione, **MVC-2**.



L'errore grande che si può fare è quello di pensare di fare una documentazione non tecnica o narrativa. Scriveremo testi basati su una sequenza di diagrammi e poche frasi di collegamento. Questo documento ha un suo linguaggio. Per componente si intende un elemento dell'architettura con delle interfacce chiare e formate. Relazioni tra più componenti espresse con delle frecce. Una **parte** è un'interfaccia esposta da una componente architetturale. Un **connettore** è una connessione tra più parti (in UML si chiama *associazione*). E' importante capire che l'architettura dev'essere valutata per la sua **qualità**. Cercheremo metodi utili e ambienti di progettazione che questi metodi importano. Questo perchè perseguiamo *qualità by construction*. Vogliamo che la **baseline** sia pulita e quindi bisogna verificare che l'architettura lo sia. La progettazione la distingueremo in due fasi (lassi di tempo contigui e continui):

- una di **alto livello** in cui fissiamo l'architettura e i pattern;

- una di **dettaglio** in cui riempiamo i buchi per agevolare il lavoro del programmatore. Progettare parti piccole che chiameremo **moduli**.

Al progettista chiediamo di garantire che la sua attività sia funzionale. Un **modulo** è un'unità di lavoro che è utile ed opportuno affidare ad un singolo programmatore. Una classe troppo grossa non va bene, sbilancia il lavoro. Voglio un insieme di caratteristiche che stiano bene insieme. Finchè non ho ridotto le classi a modulo non posso partire con la programmazione. Se ho classi molto modulari posso far lavorare più programmatori in parallelo.

Per descrivere l'architettura serve un **documento**, per produrre questo possiamo riferirci a degli standard procedurali, i quali sono però troppo rigidi. Si è capito che non è utile imporre procedure. Non è formativo imporre strutture di documenti, ma produrre processi. A noi interessa chiederci che cosa dobbiamo raccontare rispetto alle attese del committente. I documenti non devono avere una struttura standard.

Stati di processo per SEMAT:

- **Architecture selected**, dobbiamo scegliere l'architettura e dobbiamo spiegare perché essa è adatta al sistema. Selezione delle tecnologie necessarie, posso quindi fare stime di costo sensate. Decisione su **buy, build, make**. In questo stadio di avanzamento non c'è nemmeno una linea di codice, niente di realizzato;
- **Demonstrable**, dimostrazione delle principali caratteristiche del sistema agli stakeholder, i quali concordano. Decisioni su interfacce e configurazioni di sistema. Ho fatto e ho completato la progettazione ad alto livello ed eventualmente ho dei prototipi, ma non ho ancora implementato nulla;
- **Usable**, il sistema è usabile e ha le caratteristiche desiderate. Non è completamente finito, ho ancora difetti, ma essi sono accettabili. Quindi possiamo *sperare* di iniziare a fare la revisione di collaudo;
- **Ready**, il prodotto è così maturo che posso iniziare a scrivere il *manuale utente*. Si è infatti professionali se si fornisce all'utente un manuale con una struttura. E in quest'epoca non sono più solo documenti cartacei, ma anche e soprattutto con altri mezzi.

“La documentazione è l'incubo dei cowboy”. La documentazione deve essere una conseguenza dei processi organizzativi. Le attività che svolgo devono produrre tutto o quasi tutto. Serve documentare nel modo meno intrusivo possibile. Fra le cose più importanti che fa la documentazione è fornire una misura sull'avanzamento del progetto (Piano di progetto), che fornisce dei **consuntivi** (stato fotografato) e **preventivi** (stime). I consuntivi sono finali se siamo alla fine del progetto, o parziali se siamo in corso. E' bene misurare solo le cose utili e interessanti. Misureremo le cose sulle quali possiamo e vogliamo fissare degli obiettivi di miglioramento. Vorrei misurare per esempio quante volte modifico i requisiti. Misurazione per obiettivi. Il responsabile di progetto deve avere un “*cruscotto*” con gli indicatori delle metriche che utilizziamo, e questi indicatori devono essere aggiornati. Vogliamo inoltre che il responsabile spenda il meno possibile. Voglio documentare tutte le attività di pianificazione, gestione, sviluppo, verifica e validazione. In poche parole dobbiamo documentare **tutto**. Il piano ci dà degli obiettivi sui tempi e costi, le norme sono invece gli strumenti e le procedure che uso per rendere il piano fattibile. Il primo e più importante documento da realizzare (interno) sono le **norme di progetto**, che cresceranno nel tempo. Una volta che ho le norme posso pensare al piano. Questi non sono documenti di progettazione, analisi o verifica.

Ogni architettura sw ha molte viste:

- **Modello statico**, inteso come componenti, “quali sono le cose”, come si relazionano;

- **Modello dinamico**, come le parti interagiscono tra di loro;
- **Modello delle interfacce**;
- **Modello delle relazioni**, quali dati fluiscono tra i componenti;
- **Modello di distribuzione**, associazione tra nodi fisici e componenti logiche.

L'architettura va descritta in due documenti distinti:

- **Specifica tecnica**;
- **Definizione di prodotto**, ho bisogno di una *milestone* che dica che questa è una buona architettura.

Architettura della documentazione:

