Università degli studi Roma Tre

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

Tesi di laurea

Framework didattico per lo sviluppo di applicazioni per Basi di Dati

Laureando	Relatore
Vigna Federico	**Prof.Paolo Atzeni**
Matricola 224549	Università Roma Tre

Anno Accademico 2003-2004

a Rosa e Mauro: mi avete sempre sostenuto e non mi avete fatto
mai mancare nulla, insegnandomi a vivere libero.
Lasciandomi libero. A voi devo tutto;

alla nonna Pia: i tuoi angeli custodi hanno sempre studiato bene,
grazie per avermi pensato ogni momento in questi tre anni

al nonno Enea: m'hai insegnato le prime lettere dell'alfabeto
e se ora potessi leggere queste righe saresti orgoglioso di me;

ad Alessia: sei stata il mio punto di riferimento per tre anni,
accompagnandomi con tutto l'affetto.
Il mio voto sarà anche il tuo per tutti quegli esami che abbiam fatto insieme;

a Tina e Maghella perchè mi avete ridato il sorriso
tutte le volte in cui sarebbe stato facile perderlo.

Grazie a tutti i miei parenti perchè nonostante
le distanze mi siete sempre stati vicini;
a tutti i miei amici, a partire da Silvio perchè ognuno
di voi m'ha lasciato qualcosa dentro che tante volte mi ha sostenuto,
ad Ale, Nata ed Etu perchè se mi sono iscritto è anche "colpa" vostra,
a chiunque ha creduto in me.

Indice

Introduzione

La scalabilità e la riusabilità sono le basi per un software competitivo, che accolga i continui cambiamenti delle realtà informatiche. In questo ambito si inseriscono, per loro natura, i *framework*, strumenti che offrono servizi fondati sulla flessibilità e sulla portabilità.

I progettisti e gli sviluppatori di software che utilizzano linguaggi di programmazione orientati agli oggetti (Object Oriented) trovano nei framework uno strumento naturale che li astrae dai dettagli implementativi specifici, permettendogli di risparmiare tempo e di concentrarsi sui problemi, affrontandoli ad un livello più astratto.

Una delle problematiche tipiche che i framework risolvono riguarda la "difficile" relazione tra i linguaggi di programmazione O.O. e il mondo delle Basi di Dati Relazionali (RDBMS)[1].

Questo difficile colloquio, noto come *"Impedence Mismatch"* - disaccoppiamento di impedenza - ossia la differenza tra il modello relazionale e quello ad oggetti, richiede allo sviluppatore sforzi aggiuntivi per rendere possibile lo scambio di informazioni dagli oggetti del suo software al RDBMS e viceversa.

La realtà didattica incrementa questo divario. Lo studente infatti ha nozioni di programmazione O.O. in *java* ma difficilmente riesce già ad averne padronanza, specie di aspetti puramente tecnici come l'utilizzo di interfacce grafiche (es. Swing[2]) o la gestione degli eventi. Anche le conoscenze di RDBMS, in particolare

[1] del resto parlare della "immediata" relazione con le Basi di Dati ad Oggetti (ODBMS) risulta inutile vista la attuale scarsa diffusione e standardizzazione degli stessi.

[2] Pacchetto delle API di java dedicato alla gestione della grafica

per ciò che concerne le interazioni con le applicazioni in java, sono generiche e poco profonde.

L'obiettivo è quello di fornire servizi allo studente per creare applicazioni autonome e fornire un "collante" per limitare il *gap* fornito dall'*Impedence Mismatch*.

Lo studente vuole sbarazzarsi di certi dettagli implementativi e creare delle applicazioni che colloquino con l'RDBMS, scrivendo poco codice e ottenendo risultati anche "accattivanti" tramite interfacce grafiche; ne segue una gratificazione dello studente che può creare qualcosa di più dei semplici *input/output* in *console*.

Del resto anche il docente del corso è interessato affinchè lo studente si concentri sull'utilizzo dei RDBMS da java piuttosto che realizzare applicazioni perdendosi nei dettagli, e soprattutto costringendolo ad utilizzare gli strumenti tipici di connessione agli RDBMS di java (*JDBC* - v.appendice C)

Per accogliere le esigenze di studenti e docenti è stato quindi ideato, progettato ed implementato il framework BDFRAME, oggetto della tesi.

BDFRAME fornisce servizi per agevolare lo studente a superare le barriere suddette, offrendogli un *package* di elementi molto compatto e di semplice utilizzo, obbligandolo però ad implementare autonomamente il codice per il collegamento all'RDBMS.

Il framework è stato sviluppato in java e sviluppato secondo un processo *UP* - Unified Process - che consente di accogliere integrazioni e modifiche per l'aggiunta di nuovi servizi. A supporto di tale attività, sono stati utilizzati vari diagrammi UML.

In questo contesto, le metodologie, tipicamente applicate a sistemi interattivi, sono state liberamente adattate allo sviluppo di un framework.

Sono stati infine utilizzati e analizzati i package di java per l'utilizzo della grafica e degli eventi, oltre alle librerie di riflessione[3] per la gestione di alcuni

[3]Classi di java che rappresentano - *riflettono* - classi, interfacce e oggetti nella Java Virtual Machine

servizi.

La tesi presenta l'analisi e l'approfondimento delle esigenze didattiche cui si abbinano i servizi offerti dal framework, quindi mostra l'attività di sviluppo dello stesso, attraverso diagrammi UML e parti di codice maggiormente significativi.

E' così articolata:

Capitolo 1. L'approccio al corso di basi di dati. La situazione e le problematiche tipiche di uno studente che segue il corso di basi di dati.

Capitolo 2. Un framework didattico. Ruolo dei framework in generale e come un framework può aiutare un programmatore.

- **Sezione 2.1: Il framework al servizio dello studente.** Come è stato studiato BDFRAME per aiutare lo studente del corso di basi di dati.
- **Sezione 2.2: Esigenze e soluzioni.** Sono individuate le esigenze dello studente e per ognuna presentata la soluzione offerta da BDFRAME.

Capitolo 3. Lo sviluppo del framework. Diagrammi UML e codici significativi nel percorso di sviluppo del framework suddiviso per servizi offerti.

Appendici. Sono riportati a titolo informativo argomenti considerati scontati nella stesura della tesi, come il linguaggio SQL e JDBC.
Inoltre è presentata la documentazione completa (Javadoc) a corredo del framework.

Capitolo 1

L'approccio al corso di basi di dati

Per capire in che modo un framework può aiutare uno studente del corso di basi di dati è fondamentale innanzitutto calarsi nella parte per capirne le effettive esigenze.

1.1 Basi di dati

Il corso di Basi di Dati [HPBDD] mira ad insegnare vari aspetti dal linguaggio SQL (v. appendice B), sia DDL che DML, alla progettazione e realizzazione di basi di dati di media complessità.

Lo studente, di norma, non conosce ancora nessuna base di dati specifica e durante il corso gliene vengono proposte alcune di maggiore o minor complessità:

- Microsoft Access
- DB2 [DB2]
- Mimer SQL [MIMER]
- PostgresSQL [POSTGRES]

La scelta più frequente cade su Microsoft Access e normalmente anche le applicazioni di test fornite dai docenti comprendono basi di dati di questo tipo sfruttandone la portabilità.

1.2 Java e Basi di Dati

Lo studente si presenta al corso di Basi di Dati con conoscenze di Programmazione Orientata agli Oggetti e sviluppa in Java. Lo stesso infatti ha seguito già i corsi di Fondamenti di Informatica I e II e P.O.O. quindi ha una buona conoscenza di java anche se non ha avuto occasione, se non per propria iniziativa, di studiare ed utilizzare i *package* relativi alla gestione della grafica e degli eventi (*javax.swing.**, *java.awt.** [API]). Durante tali corsi ha imparato anche ad utilizzare package esterni, ossia librerie non facenti parti delle API di java.

Secondo gli obiettivi formativi del corso ci si aspetta che lo studente sappia realizzare applicazioni che utilizzino basi di dati anche di grande complessità. A tal fine vengono impartite allo studente nozioni sul collegamento tra java e i RDBMS, attraverso lo studio di Jdbc (v. appendice C), che è parte integrante del programma e sul quale vengono richieste prove di approfondimento tramite homework oltre che nella stessa prova d'esame.

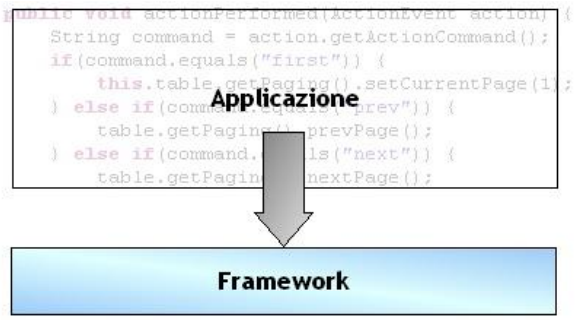
Lo studente è in grado, grazie alle lezioni e alle esercitazioni, di scrivere codice per creare autonomamente una connessione Jdbc, fare semplici interrogazioni e modifica dei dati, oltre che curare aspetti correlati come la creazione di sorgenti ODBC per il driver JdbcOdbc (v. appendice C) necessari ad esempio per raggiungere basi di dati Microsoft Access.

Capitolo 2

Un framework didattico

L'utilizzo di framework nelle attività di sviluppo di software orientato agli oggetti, accelera notevolmente i tempi permettendo ai progettisti di occuparsi dei problemi in astratto senza scendere nei dettagli - rischio sempre presente durante l'attività di progettazione.

Il framework infatti può esser visto come un livello aggiuntivo indipendente al quale chiedere servizi ignorando come è fatto è cosa sta dietro.



Come si colloca un framework

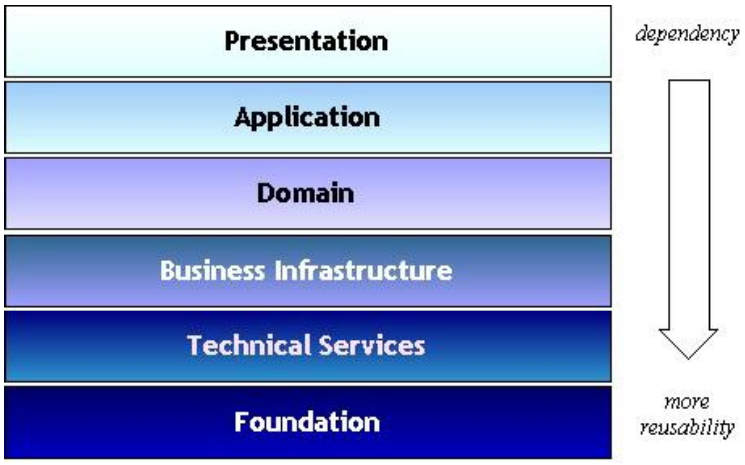
Questa struttura sposa benissimo pattern architetturali come *Layers* [BMRSS96] dove la struttura logica del sistema viene suddivisa in livelli, separando *interfacce grafiche, gestione della sessione, logica applicativa, servizi di business, servizi tecnici, fondamenta.* Questa suddivisione pone l'enfasi sul fatto che i livelli supe-

riori utilizzano e conoscono i livelli inferiori. I livelli inferiori invece non utilizzano quelli superiori, al più li conoscono solo tramite interfacce apposite.

Questo fa si che la riusabilità aumenta scendendo di livello.

I primi due livelli infatti sono dedicati alla specifica UI e, così come il terzo, fortemente accoppiate dall'applicazione specifica. Dal quarto livello si pongono livelli sempre più disaccoppiati con l'applicazione offrendo servizi di interesse generale, quindi maggiormente riusabili.

Nell'ultimo livello in particolare - *foundation* - risiedono tutti i servizi tecnici di basso livello, i framework e altri servizi di utilità. Il framework infatti, rispettando i principi di Layers, non sa nulla di chi lo utilizza - in questo caso i livelli soprastanti - mentre, al contrario, viene usato liberamente dai livelli soprastanti.

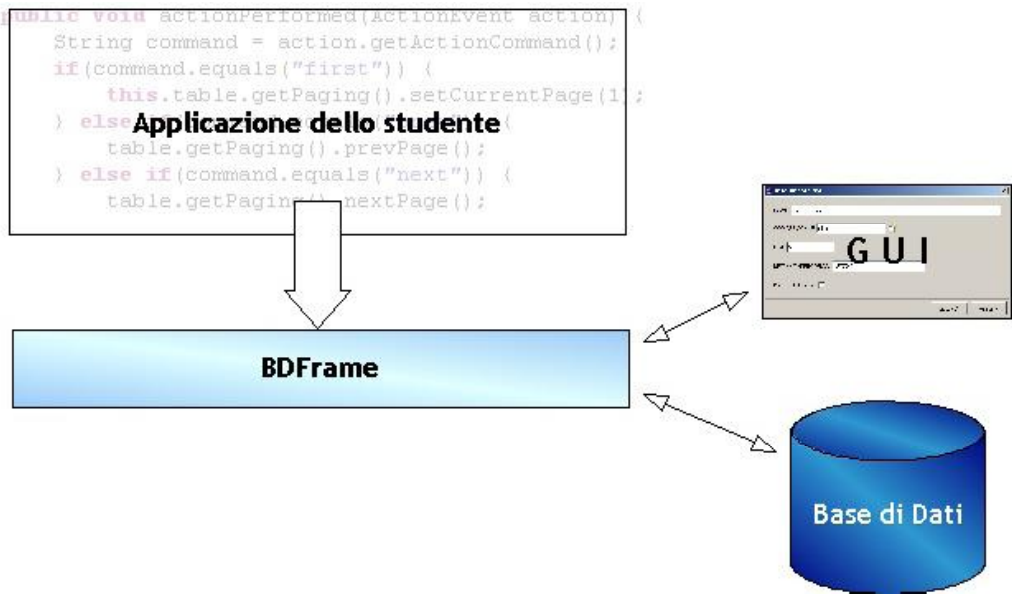


Design Pattern Layers

In genere l'utilizzo di framework, proprio per questa separazione d'interessi, porta benefici a tutto il sistema, in termini di *alta coesione* e *basso accoppiamento* (High Coesion e Low Coupling). Inoltre il framework nasconde la struttura interna fornendo solo poche classi e metodi per l'utilizzo dei servizi garantendo quindi *protezione dalle variazioni* (Protected Variation).

Nel caso specifico la necessità è di offrire un insieme di servizi allo studente, offrendo poche classi, con pochi metodi e quanto più semplici possibile.

Il framework BDFRAME si colloca proprio come uno strato software a sè stante, al quale il programmatore - negli obiettivi, uno studente - chiede servizi come fossero API di java.



Come l'applicazione dello studente interagisce con il framework

2.1 Il framework al servizio dello studente

L'obiettivo di BDFRAME è di fornire allo studente uno strumento che permetta un approccio semplificato e trasparente alla programmazione in java per l'utilizzo di Basi di Dati e la gestione di istruzioni SQL. Per esigenze di corso però lo studente deve comunque utilizzare gli strumenti tipici per l'accesso alle basi di dati in Java - jdbc [v.appendice C] - oltre ai comandi SQL.

Il framework aiuta lo studente a superare alcune barriere di Java come il non immediato approccio alle interfacce grafiche e agli eventi, in particolare offrendo oggetti semplici e intuitivi. A tal fine offre una documentazione completa (v. appendice A), nel classico stile *API di Java*.