

GESTIONE DEGLI UTENTI
IN UN'APPLICAZIONE DI BUSINESS PROCESS MANAGEMENT
PER VENETO SVILUPPO
Tesi di Laurea in Informatica

Fabio Gomiero

Università degli studi di Padova
Facoltà di Scienze MM.FF.NN.
Dipartimento di Matematica Pura ed Applicata
Corso di Laurea Triennale in Informatica

23 luglio 2009

Veneto Sviluppo è la società finanziaria della regione Veneto si occupa di:

- fornire agevolazione alle imprese venete;
- partecipare sul capitale di rischio di società.

Il mio ruolo:

si affianca alla realizzazione del nuovo gestionale (*F3000*), un'applicazione web basata sul B.P.M. per gestire le richieste di finanziamento.

Attualmente la società sta usando l'applicazione F2004.

Obiettivi dello stage

Durante la pianificazione dello stage si erano prefissati questi due obiettivi:

- 1 Realizzare un'interfaccia grafica per la gestione degli utenti.
- 2 Realizzare una seconda interfaccia grafica per la gestione dei processi e delle regole.

Entrambe le interfacce dovevano integrarsi in F3000.

Primo obiettivo

Interfaccia grafica per la gestione degli utenti

La realizzazione dell'interfaccia grafica per la gestione degli utenti doveva:

- poggiarsi su un modello dati già esistente.
- introdurmi alle logiche del B.P.M.
- farmi prendere confidenza sulla struttura e sulle tecnologie del progetto F3000.

Secondo obiettivo

Interfaccia grafica per la gestione dei processi

Doveva consentire di poter:

- creare nuovi processi di business e modificare quelli esistenti;
- aggiungere, modificare le regole coinvolte nel processo;

E' stato compiuto lo studio sul BPM ma F3000 aveva subito notevoli ritardi nel suo sviluppo.

Tali ritardi hanno compromesso lo svolgimento di quanto pianificato perchè:

- nessun modello dati per gli utenti era stato realizzato, nè tanto meno studiato.
- non c'era nessuna definizione di processo di business per implementare uno strumento per amministrarlo.

L'impossibilità di compiere gli obiettivi di partenza ha provocato una ri-pianificazione dello stage, individuando queste due azioni:

- studio e realizzazione di un modello dati per gli utenti/gruppi e permessi;
- realizzazione dell'interfaccia grafica per amministrare l'identity management;

Il modello dati da realizzare doveva:

- adattarsi il meglio possibile al processo di business;
- rispettare i permessi che erano presenti per F2004.

Queste due attività sono state compiute attraverso:

- *un difficile* reverse-engineering della base dati del vecchio gestionale;
- interviste con il tutor;
- la redazione automatizzata di report che permettessero di verificare la situazione di F2004 e di combinazioni utili da implementare per F3000.

Quest'ultima fase utilizzata anche per creare le *insert* SQL nel db di F3000.

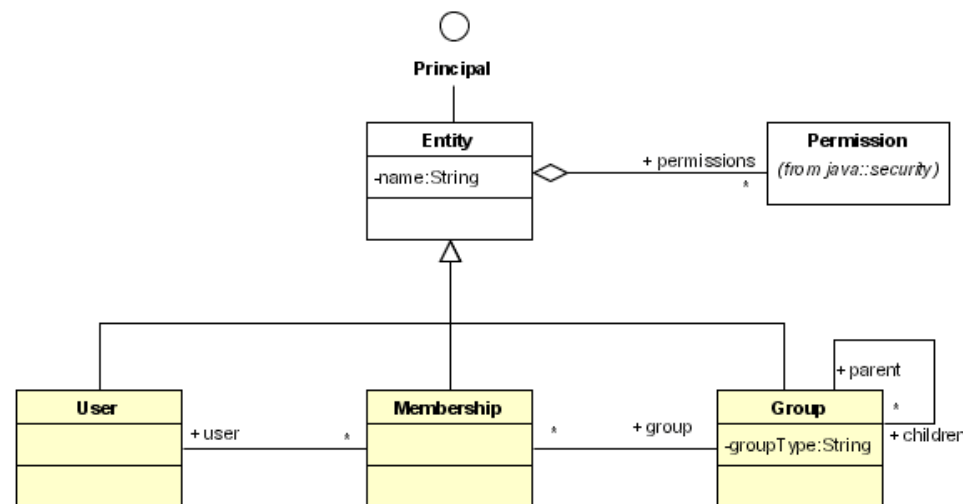
Analisi - Il modello dati

Soluzioni: Utenti e Gruppi

In base allo studio del database di F2004 e a colloqui con il tutor interno che gestisce l'applicazione ho deciso di adottare:

- utenti e gruppi strutturati secondo il modello Identity fornito dal jBPM:
 - tre tipi di Membership: Role_User, Role_Capogruppo e Modello;
 - tre tipi di gruppi: Modello, Anagrafica, TipoUtente;

⇒ mantenimento della struttura: utente-banca, banca-filiale, banca-capogruppo;



Sono stati individuati sei permessi:

- **Compito:** l'elenco di tutti i task del processo;
- **Fase:** l'elenco di tutti i task-node del processo;
- **Funzione:** l'elenco delle attività dell'applicazione che l'utente può eseguire;
- **Report:** l'elenco di tutti i report presenti nell'applicazione;
- **Tipo prodotto:** l'elenco dei prodotti che l'utente può gestire;
- **Transizione:** identifica in quale task o task-node l'utente può passare.

Ognuno di questi permessi è poi distinto in:

- **Permessi propri:** permessi assegnati specificatamente all'utente;
- **Permessi ereditati:** permesso ereditato dal gruppo;
- **Tutti i permessi:** permessi che non compaiono nelle due liste precedenti.

F2004:

- compatibile solo con il browser Internet Explorer (6+);
- amministrazione degli utenti poco intuitiva e molto dispersiva:
 - utenti visualizzati in forma tabellare;
 - permessi amministrabili in diversi punti e non facilmente comprensibili;
 - permessi ereditati dal gruppo di appartenenza non tutti modificabili;
 - passaggio da un tab dell'applicazione ad un altro non conserva le modifiche apportate se non venivano salvate.

L'applicazione ha assunto un carattere:

- prototipale;
- stand-alone perchè:
 - mancava un'infrastruttura su cui realizzare l'applicazione web;
 - altre proposte troppo complesse per chi doveva utilizzare il prodotto;

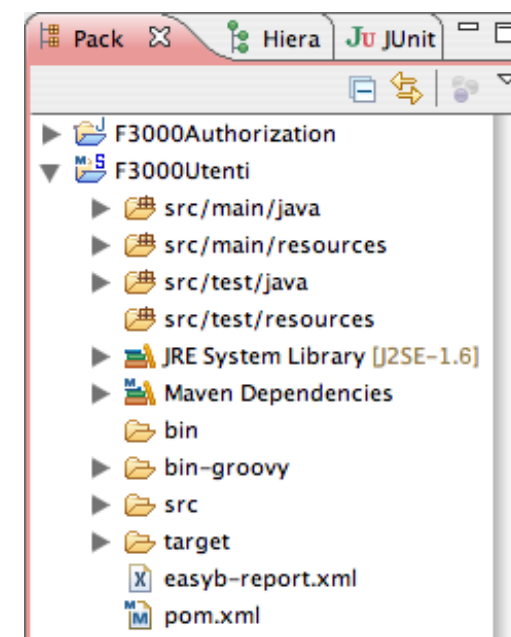
Caratteristiche:

- interfaccia grafica implementata con *javax.swing*;
- visualizzazione degli utenti *ad albero*;
- azioni di inserimento, modifica e cancellazioni permesse solo scegliendo dalla combobox *il gruppo di nome Anagrafica*;
- permessi gestiti mediante liste con *spostamento singolo o multiplo*;
- presenza di un *campo per la ricerca* dei permessi nella varie liste.

Architettura del progetto e struttura dei packages

Il progetto:

- gestito da Maven di Apache;
- basato su Spring;
- utilizza Java 1.6



L'applicazione è basata sul pattern MVC.

La creazione automatizzata dei report ha seguito il *Single Responsibility Principle*.

Per l'applicazione adottati i pattern:

- Facade;
- Factory method;
- Memento;
- DAO

Conclusioni

- Tutte le scelte prese in **piena autonomia**;
- Cercato di favorire il **riuso**;
- Seguito i principi dell'**Ingegneria del Software**.
- Tutto il progetto soggetto a diversi **cambiamenti di requisiti in corso d'opera**:
 - ⇒ carattere prototipale per studiare tutti gli aspetti desiderati da Veneto Sviluppo;

Solo in questo periodo sarebbe possibile attivare uno stage con la mia pianificazione.