

Projet multimodule - GTD

Clémentine BRUN Brendan LE NY Myrtille PONGE Brian
RAVENET

Université de Nantes

3 février 2010

Qu'est-ce que GTD ?

Getting Things Done, the art of stress-free productivity, David Allen
Méthode de gestion des priorités quotidiennes.

Les tâches à réaliser sont vues selon deux axes : par contexte ou par projet.

Contexte environnement de travail (outils et lieu)

Projet objectif à atteindre (Passer un examen)

Mettre un diag ici.

Le projet multi-module

Fonctionnalités :

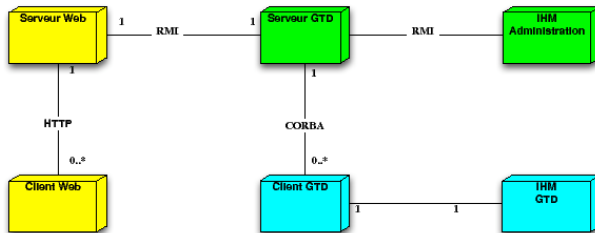
- Collecter
- Organiser
- Agir
- Revoir

Le projet multi-module

Fonctionnalités :

- Collecter
- Organiser
- Agir
- Revoir

Architecture globale de l'application :



Conception globale de l'application

Divers objectifs :

- séparation des composants pour remplacement
- possibilité de développement séparé
- séparation des préoccupations

Solutions

- Design par interface (stubs)
- Séparation en couches (n-tiers)
- MVC
- développement séparé mais nécessité d'un refactoring



Bilan

- conception fonctionnelle
- refactoring/fusion sans problème

La persistance locale

Objectifs :

- Persistence des données.
- Pas de déploiement.
- Séparation du modèle et de la persistance.
- Factorisation du code.

La persistance locale

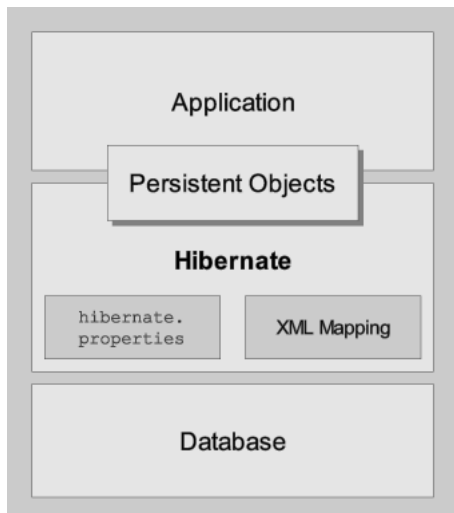
Objectifs :

- Persistance des données.
- Pas de déploiement.
- Séparation du modèle et de la persistance.
- Factorisation du code.

Les solutions :

- Une base de données embarquée : H2.
- Un framework : Hibernate.
- Une architecture en couches.
- Le pattern DAO.

Hibernate



- Accès à la BD par des appels objets.
- Configuration spécifique pour la BD.
- Deux modes : annotations ou mapping XML.

Hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration
    PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-
        configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="dialect" >org.hibernate.dialect.
            H2Dialect</property>
        <property name="show_sql" >true</property>
        <property name="generate_statistics" >true</property>
        <property name="hbm2ddl.auto" >update</property>
        <property name="jdbc.batch_size" >1</property>
        <property name="connection.driver_class" >org.h2.
            Driver</property>
        <property name="connection.url" >jdbc:h2:~/Essai</
            property>
        <property name="connection.username" >sa</property>
        <property name="connection.password" ></property>
    </session-factory>
</hibernate-configuration>
```

```

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate_
Mapping_DTD_3.0//EN" "http://hibernate.sourceforge.net/
hibernate-mapping-3.0.dtd">
<hibernate-mapping auto-import="true" package="fr.univnantes
.alma.gtd.model.gestionnaire.ressources">
  <class name="fr.univnantes.alma.gtd.model.
    gestionnaire.ressources.Idee" >
    <id name="id" column="ideeld" type="java.lang.
      Integer">
      <generator class="increment"/>
    </id>
    <property name="nom" type="java.lang.String"/>
    <property name="description" type="java.lang.String"
      />
  </class>
</hibernate-mapping>

```

```

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate_
Mapping_DTD_3.0//EN" "http://hibernate.sourceforge.net/
hibernate-mapping-3.0.dtd">
<hibernate-mapping auto-import="true" package="fr.univnantes
.alma.gtd.model.gestionnaire.ressources">
  <class name="fr.univnantes.alma.gtd.model.
    gestionnaire.ressources.Idee" >
    <id name="id" column="ideeld" type="java.lang.
      Integer">
      <generator class="increment"/>
    </id>
    <property name="nom" type="java.lang.String"/>
    <property name="description" type="java.lang.String"
      />
  </class>
</hibernate-mapping>

```

Ajout dans Hibernate.cfg.xml

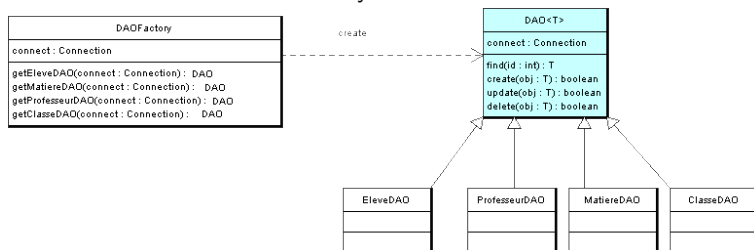
```

<mapping resource="hibernate/entities/Idee.hbm.xml" />

```

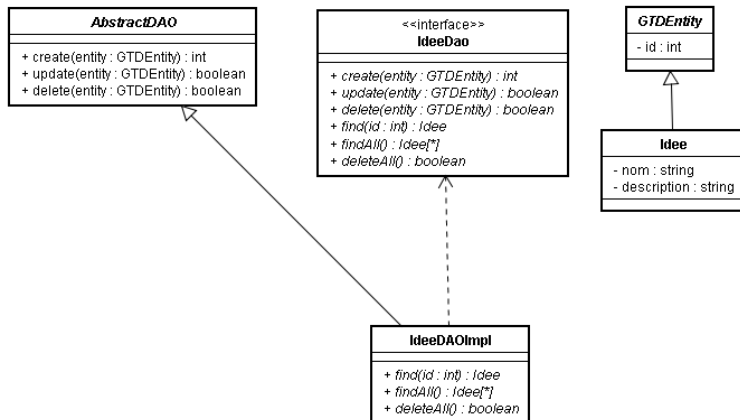
Le pattern DAO

Pattern basé sur Pattern Factory.



Le pattern DAO

TopCased : problème de templates !



La base de données

Creation á la volée par Hibernate.

Problèmes et solutions :

- Une table par hiérarchie d'héritage (ex : Pattern Etat).
- Les relations one-to-one.

Persistance distante, source de complexité

Problèmes rencontrés :

- Mauvaise évaluation du temps.
- Mauvaise synchronisation avec les autres groupes.
- Problème de cohésion des modèles.

Conséquence sur GTD :

- Fonctionnalités non implémentées dans GTD.

Solution de persistance sur un serveur distant via CORBA

- Implémentation de l'interface IPersistance utilisée dans la persistance locale. Voici les fonctions que nous aurions implémentées pour chaque élément persistant :

```
public List<Idee> findAllIdees();  
public Idee findIdee(final Integer id);  
public Boolean update(final Idee entity);  
public Boolean deleteAllIdees();  
public Boolean delete(final Idee entity);
```

Solution de persistance sur un serveur distant via CORBA

- Patron de conception proxy pour définir l'ensemble de classe représentant les classes du modèle. Les instances de ces classes seraient des représentations des objets corba distants.
- Ces classes contiendraient un IOR permettant de trouver l'objet distant.
- Regroupement des opérations serait dans une classe unique, seul le type d'objet que l'on fait persister varie.
- Paramétrage de cette classe unique par un type T représentant le type d'objet à persister.

Signature des classes :

```
abstract class ObjetDistant<T extends GTDEntity>  
class Tachempl extends ObjetDistant<Tache> implements  
    Tache
```

- Pour communiquer avec le serveur nous aurions besoin de l'adresse du registre CORBA qui sera dans un attribut.

De l'UML au code source...

- volonté de fructifier le travail de modélisation
- intégration du modèle dans le processus de développement (MDA)

Écriture de scripts Acceleo

Implémenté :

- interfaces
- classes, héritages et réalisations
- attributs internes et relations (cardinalités)
- méthodes internes, setters, getters, contractuelles (visibilités)

Potentiellement :

- annotations et fichiers de configuration JPA
- classes stéréotypées DAO, Singleton, Etat

Bilan

- langage difficile à adopter (API de réflexion ?)
- pérenité du modèle ?
- pertinent pour les classes données, et le reste ?
- + gain de temps sur le long terme (pérénisation des scripts)
- + cohérence modèle-code préservée
- + plus grande agilité du modèle

IHM : Conception et intégration

- Deux problématiques
- La réalisation des accès aux fonctionnalités de la méthode GTD
- Une conception centrée utilisateur et ergonomique

Les choix d'implémentation, architecture logicielle

- Notre application étant développée en java, nous avons décidé d'utiliser la librairie SWING utilisée en TP d'IHM.
- Nous avons développé l'IHM selon le patron de conception MVC.
- Séparation du Modèle, de la Vue et du Contrôleur (spécification des composants, conception des interfaces)
- Pour pouvoir développer l'IHM en parallèle de l'application nous avons réalisé des bouchons pour simuler le comportement du contrôleur.

Les points mis en valeur dans notre IHM, critères ergonomiques

Guidage de l'utilisateur : Groupement entre informations relatives au panier, au contexte ou au projet.

Densité informationnelle réduite grâce à l'utilisation d'onglet.

Contrôle explicite : message d'information en cas de mauvaise utilisation des fonctionnalités

Codes simples : "+" pour ajout.

Homogénéité, cohérence

Correction des erreurs

Le compromis dans la réalisation

- Vision des concepts GTD dans l'interface
- Accès rapides aux fonctionnalités

Gestion de projet

Une équipe de quatre personnes :

- + Partage des tâches.
- + Outils collaboratifs (SVN, Google Document).
 - Définition d'un calendrier (diagramme de Gantt).

Gestion de projet

Une équipe de quatre personnes :

- + Partage des tâches.
- + Outils collaboratifs (SVN, Google Document).
 - Définition d'un calendrier (diagramme de Gantt).

Des interactions avec d'autres équipes :

- + Partage d'informations.
- Prise de décisions.

Un projet innovant :

- Les technologies utilisées.
- L'organisation du projet.
- La taille du projet.

Un projet innovant :

- Les technologies utilisées.
- L'organisation du projet.
- La taille du projet.

Un projet réaliste ?