

Conclusion générale

Il y a un moment où il faut se retourner et de revoir le chemin qu'on a fait pour pouvoir construire le futur. Le *design pattern* représente une partie de notre expérience et de notre solution ; nous devrions l'exploiter pour construire des outils et des méthodologies pour avancer vers la maîtrise des systèmes qui nous entourent.

La technologie sur la modélisation de système informatique est un domaine de l'abstrait qui se base sur des définitions et des représentations. Il est parfois difficile de prouver une théorie ou une spécification sans l'instanciation. Mais avec les patterns, on n'a pas besoin de démontrer une solution car on suppose déjà qu'ils sont prouvés.

La modélisation d'un système d'information avec des *design patterns* est donc une conception logique et qui devrait constituer la base de la nouvelle génération de génie logiciel.

La transformation de l'information par le processus métier et l'interprétation des données par le processus décisionnel constitue le moteur du système d'information ; avec ces deux blocs principaux, on peut en déduire que l'outil de gestion de système d'information est composé de :

- gestionnaire de processus,
- gestionnaire de ressources,
- gestionnaire des services,
- générateur de rapport,
- organe de communication des indicateurs

Ces outils de base tournent dans une architecture modulaire et distribuée qui est commandable par les acteurs du système à partir d'un seul noyau fixe que l'on a appelé **Q-BPM**.

On a proposé cette étude dans le cadre de formalisation des modèles de conception et d'apporter notre vision du système d'information. Elle nous a permis d'identifier les différents composants techniques qui constituent le SI et leurs interactions. En plus on est arrivé à identifier les différentes classes de l'information dans le SI au cours de son traitement :

- information physique
- information codée
- information de masse
- information agrégée
- information clé
- incident
- risque
- décision
- objective

Ces types de réorganisation de l'information peuvent prendre plusieurs états et plusieurs interprétations selon le contexte et le niveau de traitement. Avec cette complexité de la notion de l'information, le SI reste encore un système complexe et ouvert, et on a besoin de continuer la recherche pour augmenter notre maîtrise du système et notre savoir-faire sur l'information.

ANNEXES

ANNEXE 1 : Framework Ressource

Ce Framework fait partie des piliers du système Quanta. Elle fournit des services élémentaires pour créer ou accéder à une ressource.

A1.1 Architecture

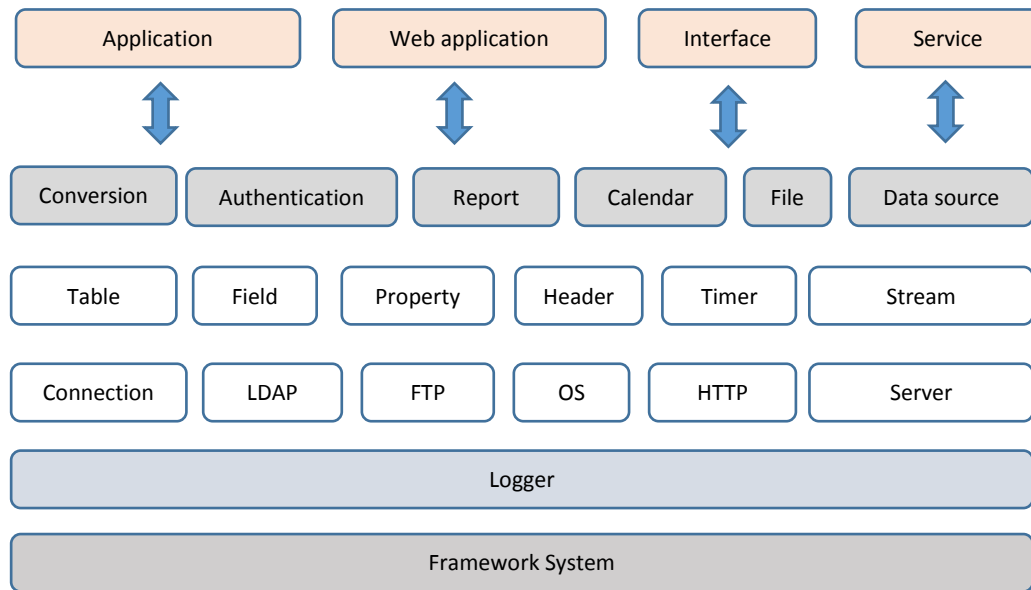


Figure 7.1 : Couches logicielles du Framework ressource

Le code relatif aux implémentations des classes dépend entièrement des Framework offerts par l'environnement de développement.

L'architecture est composée de quatre couches :

- couche log : pour tracer les traitements afin de faciliter la maintenance
- couche accès aux ressources
- couche modèle de données
- couche de traitement et d'interface qui est en relation direct avec d'autre système extérieur

A1.2 Package

Tableau 7.1 : Liste des packages

Package	Description
com.hitechcommons.resources	Contient les modèles de données
com.hitechcommons.resources.component	Contient des composants graphiques lourds
com.hitechcommons.resources.component.calendar	Contient des composants calendrier
com.hitechcommons.resources.component.grid	Contient des composants tableaux
com.hitechcommons.resources.conversion	Contient des services de conversion de type
com.hitechcommons.resources.conversion.loggers	Contient de classe d'enregistrement de Log
com.hitechcommons.resources.database	Contient de modèle de données enregistrement
com.hitechcommons.resources.database.gdb	Contient des composants de connexion aux bases de données
com.hitechcommons.resources.file	Contient des modèles de données flux
com.hitechcommons.resources.office	Contient des composants de gestion de fichier office
com.hitechcommons.resources.office.open	Contient des composants de gestion de fichier open office
com.hitechcommons.resources.property	Contient des modèles de données propriétés : liste des caractères d'un objet
com.hitechcommons.resources.service	Contient des interfaces d'accès aux ressources
com.hitechcommons.utils	Contient des composants de gestion de ressources système

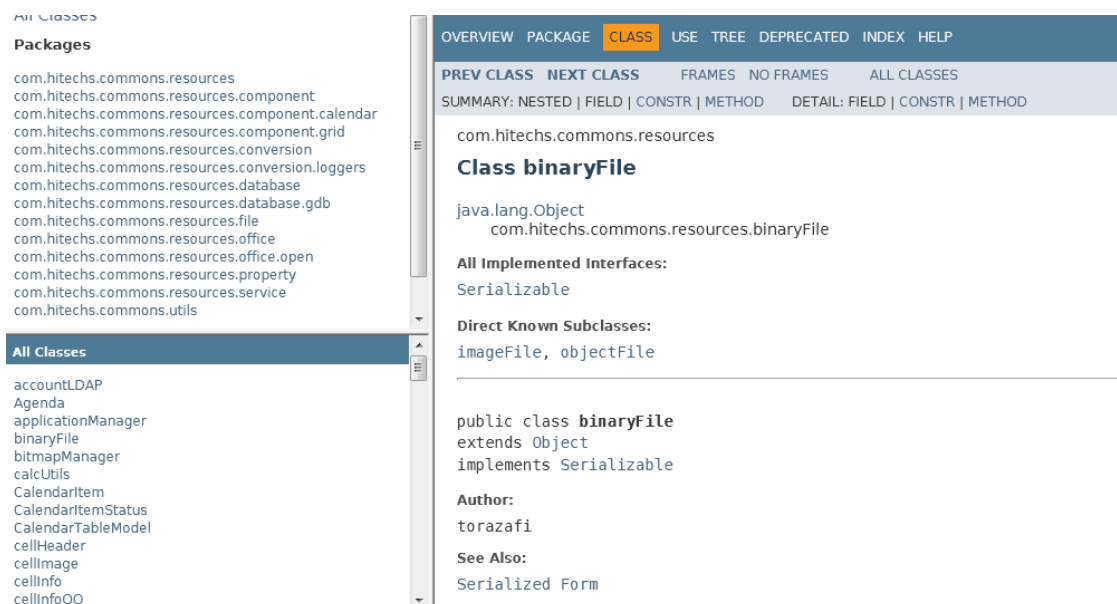


Figure 7.2 : Java doc du Framework

A1.2.1 Package resources

Tableau 7.2 : Liste des classes ressources

Class	Description
accountLDAP	Modèle de donnée LDAP
binaryFile	Modèle de donnée binaire
componentUni	Classe mère de composant
constanteUni	Contient des constantes
dataCron	Modèle de données Cron
dataForm	Modèle de données formulaire
dataMail	Modèle de donnée Mail
dataObject	Modèle de donnée liste
dataUni	Modèle de donnée enregistrement
field	Specify a field and his default value *
fieldUni	Modèle d'un champ
ftpServer	Modèle de données d'accès au serveur FTP
headerUni	Modèle de données d'entête
imageFile	Modèle de données Image
ldapServer	Modèle de données d'accès au serveur LDAP

objectFile	Modèle de flux d'objet
pageUni	Modèle de données pour gérer une page
pointUni	Modèle de données Point
proxyData	Modèle de données Proxy
rectangleUni	Modèle de données Rectangle
tableUni	Modèle de données Table
textFile	Modèle de données de flux texte
vectorUni	Modèle de données liste indexé

A1.2.2 Package component

Tableau 7.3 : Liste de classes composantes

Class	Description
imageView	Classe de traitement d'image
pageComponentUni	Classe de gestion de page
panelImage	Classe d'affichage d'image

A1.2.3 Package calendar

Tableau 7.4 : Liste des classes calendriers

Class	Description
Agenda	Modèle de données Agenda
CalendarItem	Modèle de données Item de calendrier
CalendarTableModel	Modèle de données liste des items
DayPanel	Classe d'affichage de Jour
DayView	Classe d'affichage de Jour avec format
MonthView	Classe d'affichage de mois
View	Classe d'affichage de calendrier
WeekView	Classe d'affichage de week

A1.2.4 Package grid

Tableau 7.5 : Liste des classes tableau

Class	Description
dateEditor	Modèle d’affichage de date
gridExport	Classe de transformation de données
gridUni	Modèle de donnée du tableau
metaColumn	Modèle de colonne
paneGrid	Classe d’affichage
paneGridPage	Classe d’affichage avec page
SpinnerEditor	Modèle d’affichage de chiffre

A1.2.5 Package conversion

Tableau 7.6 : Liste de classe de conversion

Class	Description
applicationManager	Gestionnaire du runtime Java
bitmapManager	Gestionnaire d’image
controlConnection	Gestionnaire de connexion oracle
dataComparator	Classe de comparaison
dateConversion	Gestionnaire de date
fileManager	Gestionnaire de fichier
ftpManager	Gestionnaire de connexion FTP
httpManager	Gestionnaire de connexion HTTP
jarManager	Classe de manipulation de fichier jar
ldapManager	Gestionnaire de connexion LDAP
mailManager	Gestionnaire de connexion au serveur Mail
nativeConversion	Classe de conversion des données natives
PDFManager	Classe de manipulation de fichier PDF
privilegedAction	Gestionnaire d’accès aux fichiers
requestParser	Parseur des objets Request

splashImage	Gestionnaire d'image de présentation
timerUni	Classe de donnée Timer
traductorDate	Classe de donnée pour traduire une date
traductorValue	Classe de donnée pour traduire objet
utilsData	Gestionnaire des données enregistrements
utilsForm	Gestionnaire de données formulaires

A1.2.6 Package loggers

Tableau 7.8 : Liste de classe d'enregistrement d'événement

Class	Description
ConnectionLogFactory	Constructeur d'entité Log
logConfigurationFactory	Classe de données de configuration
logJDBCConfigurationFactory	Classe de données de configuration pour une base de données
logWriter	Gestionnaire d'enregistrement de Log

A1.2.7 Package database

Tableau 7.9 : Liste de classe de gestion des données

Class	Description
comparatorUni	Classe de comparaison
connection	Classe de connexion
connectionManager	Gestionnaire de liste de connexion
connectionParameter	Classe de données de configuration d'accès
orderDataTable	Classe de données pour gérer un tri
queryBuilder	Classe de création de requête SQL
queryUtil	Classe de traitement de requête SQL
whereDataTable	Classe de données pour gérer le clause where d'une requête SQL

A1.2.8 Package gdb

Tableau 7.10 : Liste de classe de gestion des connexions

Class	Description
connectionMySQL	Implémentation d'une connexion Mysql
connectionOdbc	Implémentation d'une connexion ODBC
connectionOracle	Implémentation d'une connexion Oracle
connectionPostgres	Implémentation d'une connexion PostgreSQL
connectionSqlServer	Implémentation d'une connexion Sql Server
metaMysql	Classe d'entête de connexion Mysql
metaOdbc	Classe d'entête de connexion ODBC
metaOracle	Classe d'entête de connexion Oracle
metaPostgres	Classe d'entête de connexion PostgreSQL
metaSqlServer	Classe d'entête de connexion SQL server

A1.2.9 Package office

Tableau 7.11 : Liste de classe de traitement de fichier office

Class	Description
cellHeader	Classe de donnée de cellule
cellImage	Classe de donnée image
cellInfo	Classe de données sur le contenu de cellule
contentSheet	Classe de données d'une feuille
contentWord	Classe de donnée d'un document
pdfSheet	Convertisseur de feuille en PDF
pdfWord	Classe de conversion de document en PDF
rangeInfo	Classe de donnée pour un range de cellule
sheetManager	Gestionnaire de document Excel
writerHeadFooter	Gestionnaire d'entête d'un document
writerManager	Gestionnaire de document Word

A1.2.10 Package file

Tableau 7.12 : Liste de classe de données flux

Class	Description
fileInfo	Classe de données d'entête de fichier
fileResource	Classe de données fichier

A1.2.11 Package open

Tableau 7.13 : Liste de classe de traitement de fichier open office

Class	Description
calcUtils	Utilitaire sur le traitement de cellule
cellInfoOO	Classe de données cellule
connectManager	Gestionnaire de connexion avec l'application mère
constanteOo	Liste de constante
printListner	Ecouleur utilisé pour imprimer le fichier
sheets	Gestionnaire de document Sheet
tableInfo	Classe de données d'un tableau
tablesInfo	Classe de données pour une liste de tableaux
writer	Gestionnaire de document Writer

A1.2.12 Package property

Tableau 7.14 : Liste de classe de gestion des propriétés

Class	Description
groupUIProperty	Groupe de propriétés
UIProperty	Classe de donnée pour une propriété

A1.2.13 Package service

Tableau 7.15 : Classe de gestion de service

Class	Description
ResourceService	Gestionnaire de version des services ressources

A1.2.14 Package utils

Tableau 7.16 : Liste de classe réservée pour Java

Class	Description
javaRuntime	Classe utilitaire pour accéder aux ressources de Runtime Java en cours
keyDispatcher	Utilitaire Java pour partager les événements des touches claviers
listnerObjectEvent	Classe de gestion des événements

A1.3 Interface

Elle définit l'ensemble des traitements ouverts pour accéder et réutiliser l'ensemble des services offerts par le Framework.

A1.3.1 Package database

Tableau 7.17 : Liste d'interface dédié aux bases de données

Interface
ICconnection
<pre>package com.hitechcommons.resources.database; import com.hitechcommons.resources.dataObject; import com.hitechcommons.resources.fieldUni; import com.hitechcommons.resources.tableUni; import java.sql.Connection; import java.sql.PreparedStatement; import java.util.Date; import java.util.List; public interface ICconnection { String getName(); void setName(String name); short getTypeConnection();</pre>

```

String getUrlDriver();

void setUrlDriver(String urlDriver);

String getDriverName();

void setDriverName(String driverName);

Connection getConnection();

void setConnection(Connection con);

connectionParameter getConnectionParameter();

void setConnectionParameter(connectionParameter parameter);

int getIndex();

void setIndex(int index);

void openConnection();

void openConnection(String url,String login,String pwd);

boolean isClosed();

void closeConnection();

void startTransaction();

void commitTransaction();

void rollBackTransaction();

Date getDate();

PreparedStatement getStatement(String sql)throws Exception;

void addDataAuto(dataObject data);

void execute(String sql,List<fieldUni> values);

Object executeScalar(String sql,List<fieldUni> values);

List<dataObject> executeQuery (tableUni table,String sql,List<fieldUni> values);

public dataObject executeQuerySingle (tableUni table,String sql,List<fieldUni> values);

void executeProcedure(String procedureName,List<fieldUni> values);

```

```

    Object executeProcedureScalar(String procedureName,List<fieldUni> values);

    List<dataObject> executeProcedureQuery(tableUni table,String
    procedureName,List<fieldUni> values);

    List<dataObject> executeQueryListPages(tableUni table,String whereClause,String
    orderClause,int indexBegin, int plage);

    void addData(dataObject data);

    void updateData(dataObject data);

    void deleteData(dataObject data);

    Object executeScalarExtra(tableUni table, String sql);

    List<dataObject> executeQueryListExtra(tableUni table, String WhereClause,
    List<orderDataTable> orderClause);

    List<dataObject> executeQueryListPagesExtra(tableUni table, String WhereClause,
    List<orderDataTable> orderClause, int indexBegin, int plage);

    String getValueSql(fieldUni field,Object obj);

    String getValueOfWhere(fieldUni field, Object value);

    IMetaData getMeta();
}

```

IMetaData

```

package com.hitechs.common.resources.database;
import com.hitechs.common.resources.fieldUni;
import com.hitechs.common.resources.tableUni;
import java.util.List;

public interface IMetaData {

    tableUni getTable(IConnection con,String table);

    List<tableUni> getListTable(IConnection con);

    List<fieldUni> getListFields(IConnection con,String table);

    String createTable(tableUni table);

    String dropTable(tableUni table);

}

```