

TD 1 - Mise en œuvre de Design Patterns

Sylvie Vignes, Etienne Borde

Sujet : Modéliser un logiciel d'édition de documents

L'objectif est de modéliser un logiciel d'édition de document se basant sur une **représentation générique** du contenu d'un document : à partir de cette représentation, on peut alors produire une page html, un document XML, un document LaTeX ou un simple document textuel. Pour concevoir le diagramme de classes, on s'appuie sur différents DP.

1-Structuration du document

Un document contient des éléments simples (texte et image) et des éléments structurés tels que des sections, des tableaux, des listes. On peut représenter un document sous forme d'arbre dont les nœuds peuvent être des tableaux, listes, sections et les feuilles des images ou du texte. Ainsi on peut par exemple encapsuler dans un tableau une liste qui peut avoir soit une autre liste, soit une image. Le document est caractérisé par des *méta données* telles que titre et auteur. Le document est associé à la racine de l'arbre. Appliquer le pattern **Composite**

2-Mise en forme du document

Des mises en forme ou des effets peuvent être appliqués aux éléments du document. Chacun « décore » un élément de l'arbre. Par exemple, taille de la police, souligné gras, lien, bordure. Sans modifier les classes qui définissent vos éléments, utiliser le pattern **Décorateur** afin d'appliquer des mises en forme sur ces derniers. Un décorateur peut être appliqué à un élément déjà décoré. Par exemple cumuler un effet gras et un effet souligné sur un texte, une bordure. Les décorateurs sont génériques pour tous les éléments de l'arbre. Si un décorateur ne peut pas s'appliquer, par ex italique sur image, son action est simplement ignorée lors de la mise en forme.

3- Création de document à partir de templates

On voudrait créer des documents à partir de modèles prédéfinis par exemple une lettre, un CV, un rapport.

On va utiliser une variante simplificatrice du pattern **Fabrique**. Nous sommes dans le cas particulier où la seule distinction entre des instances de document concerne son contenu uniquement ; Il n'y a donc pas intérêt à créer des sous-classes de document pour chaque version. On demande d'appliquer la fabrique pour créer une instance de document en initialisant avec un arbre prédéfini.

4-Export du document dans un format concret

A partir de la modélisation de votre document, on souhaite générer des représentations en html, XML ou autre. Le pattern à utiliser est le **Visiteur**. On souhaite éviter la modification des classes existantes à chaque définition d'un nouveau format.

Valider votre conception de ce Design Pattern Visiteur par des diagrammes de séquences illustrant des scénarios.

5-Affichage multi vues (niveau avancé)

L'utilisateur du logiciel veut pouvoir afficher le document simultanément dans plusieurs formats. Dès qu'il y a une modification faite dans le document, les vues doivent être mises à jour (les visiteurs revisitent l'arbre). Pour modéliser cette fonctionnalité, adapter le pattern **Observateur**.

Références DP :

<http://rpouiller.developpez.com/tutoriel/java/design-patterns-gang-of-four/>

http://en.wikibooks.org/wiki/Computer_Science_Design_Patterns