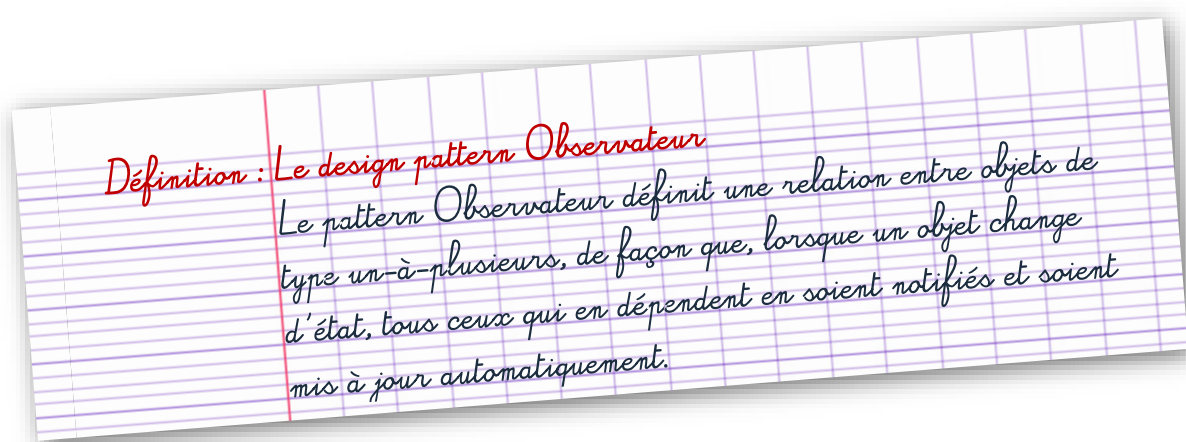


# TD 13 – Premiers Principes de Conceptions Orientée Objet

## Cours 2 Le pattern Observateur

Sans vous en rendre compte, nous avons appliqué le pattern Observateur avec la structure de classes **EventManager** / **EventListener**. Ce pattern permet de réaliser une relation un-à-plusieurs entre objets. Un **EventManager** est relié à plusieurs **EventListener**. Lorsque l'**EventManager** est modifié, il notifie chacun de ses **EventListener** qui doivent alors se mettre à jour.

### 2.1 Définition



Concrètement, son diagramme de classe est représenté sur la figure 2.

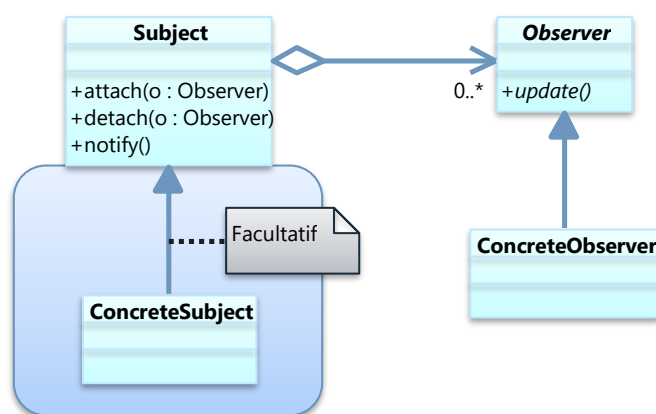


Figure 2 : Design pattern Observateur

Ce pattern est très utilisé dans les applications ayant une interface graphique. En effet, les données de l'application sont stockées dans des objets qui sont observés par l'interface graphique. Par exemple, des données représentant l'évolution temporelle d'un indicateur peuvent être affichées (observées) dans un tableau ou dans un graphique. La modification des données (événement extérieur) est alors automatiquement répercutée à l'affichage dans tous les observateurs (le tableau et le graphique).

Le pattern observateur est le moteur du pattern Modèle-Vue-Contrôleur qui est à la base de la plupart des applications graphiques actuelles (notamment les sites web)

### Cours 3 Le pattern Modèle-Vue-Contrôleur (MVC)

Ce pattern architecture les applications autour de trois types de classes

1. Le **modèle** : c'est lui qui contient les données de l'application et les traite.
2. Les **vues** : ce sont les fenêtres graphiques qui proposent l'affichage et l'interaction avec l'utilisateur
3. Les **contrôleurs** : ce sont eux qui reçoivent les actions des vues et se chargent de les traduire et éventuellement de les transmettre au modèle. Par exemple la modification d'une valeur dans un champ de texte est validé par le contrôleur puis transmis au modèle adéquat afin d'être mis à jour. Les contrôleurs sont chargés de fournir une réponse aux événements de la vue. Ainsi, en découplant les contrôleurs des vues, il est possible de fournir des réponses différentes pour les mêmes actions dans les vues (un contrôleur verbeux qui affiche des pop-up de confirmations tout le temps, ou un contrôleur silencieux, ou un contrôleur limité pour une version d'évaluation...).

Le pattern MVC implémente le pattern Observateur entre le modèle (le sujet observable) et les vues (les observateurs). Ainsi, toute modification du modèle est automatiquement répercutée à l'affichage.

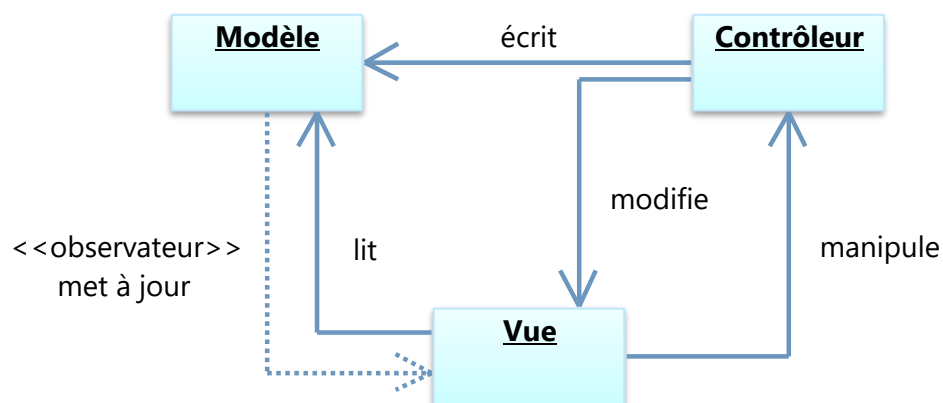


Figure 3 : Diagramme de principe du pattern MVC