

# LE PATTERN OBSERVATEUR

## Intention

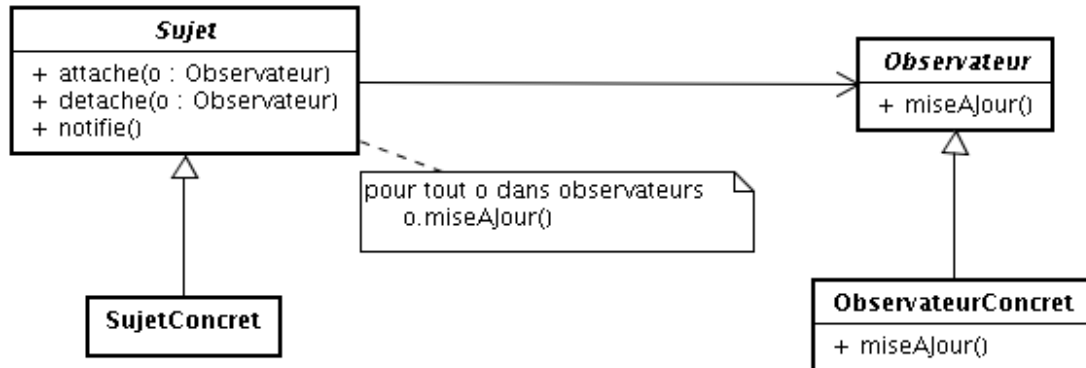
Définit une interdépendance de type un à plusieurs, de façon telle que quand un objet change d'état, tous ceux qui en dépendent en soient notifiés et automatiquement mis à jour.

## Indications d'utilisation

On utilisera le pattern observateur lorsque :

- Un concept a deux représentations, l'une dépendant de l'autre. Encapsuler ces deux représentations dans des objets distincts permet de les réutiliser et de les modifier indépendamment.
- La modification d'un objet nécessite de modifier les autres, et que l'on ne sait pas combien sont ces autres.
- Un objet doit être capable de faire une notification à d'autres objets sans faire d'hypothèses sur la nature de ces objets. En d'autres termes, quand ces objets ne doivent pas être trop fortement couplés.

## Structure



## Constituants

### *Sujet*

- Il connaît ses observateurs. Un nombre quelconque d'observateurs peut observer un sujet.
- Il fournit une interface pour attacher et détacher les objets observateurs.

### *Observateur*

- Il définit une interface de mise à jour pour les objets qui doivent être notifiés de changements dans un sujet.

### *SujetConcret*

- Il envoie une notification à ses observateurs lorsqu'il change d'état.

### *ObservateurConcret*

- Il fait l'implémentation de l'interface de mise à jour de l'observateur pour conserver la cohérence de son état avec le sujet.

## Collaborations

Le **SujetConcret** notifie ses observateurs de tout changement se produisant qui pourraient rendre l'état de ses observateurs incompatible avec le sien propre.

Après avoir été informé d'un changement dans le sujet concret, un objet **ObservateurConcret** met son propre état en conformité avec celui du **SujetConcret**.