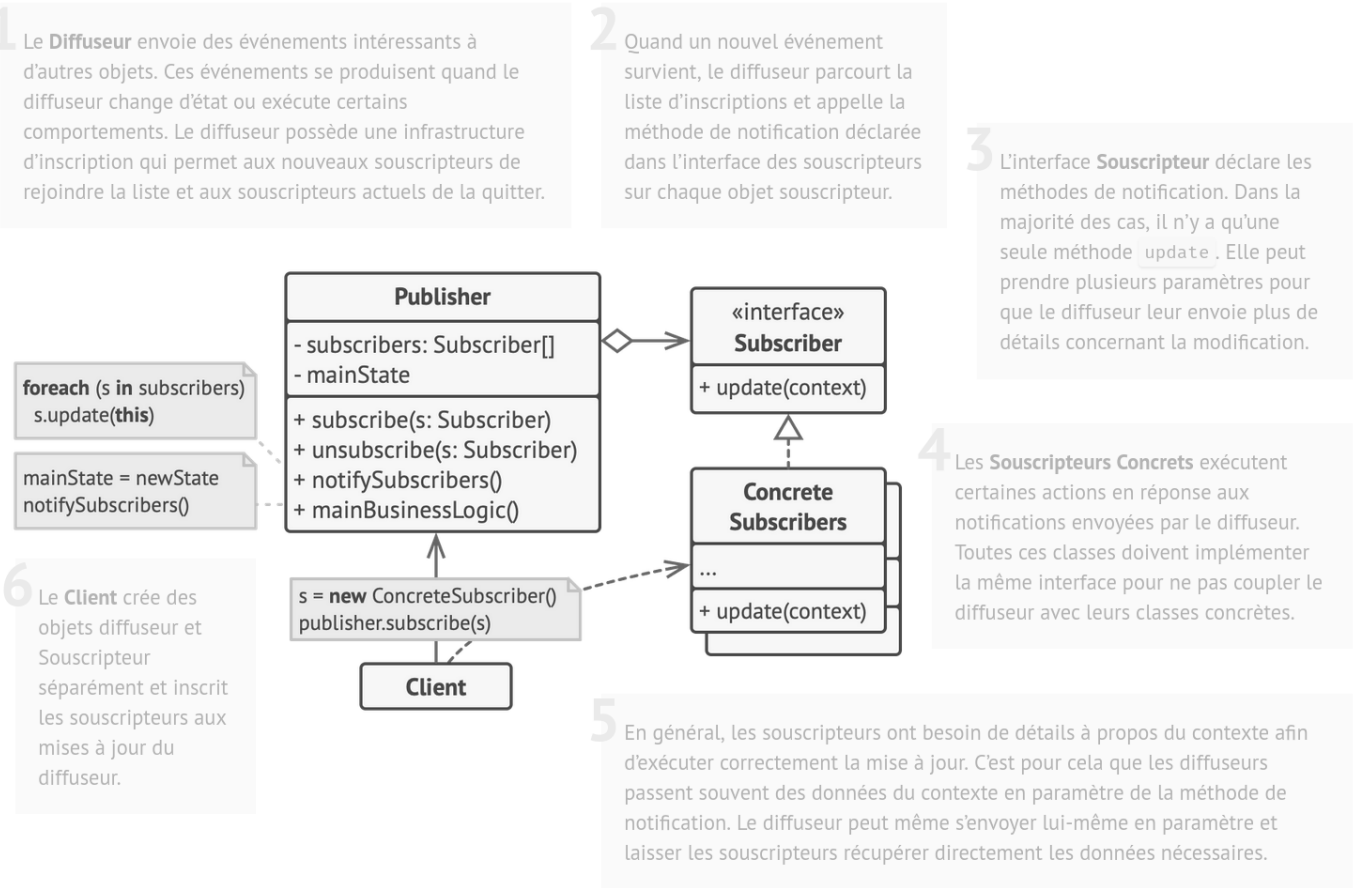
Les observables

Problème

Lorsque l'on a des évènements (par exemple nouvel articles) et que l'on a plusieurs terminaux pour recevoir cet evènement (écrans, notifications, emails, etc). Si on fait simplement tout dans une seule classe, cela enfreint le principe d **ouvert-fermé** (c'est à dire qu'un module doit être ouvert au changements sans avoir à le modifier).

Car ici pour ajouter ou supprimer des terminaux (observateurs), on est obligé de modifier tout le module.

Solution



La classe qui publie les évènements va stoquer une liste d'observateurs et va implémenter des méthodes pour que les observateurs puissent s'abonner ou se désabonner. Les observateurs vont implémenter l'interface "Obervateur" qui va simplement contenir une méthode pour recevoir l'évènement. A present pour ajouter ou supprimer des terminaux il suffit d'abonner ou désabonner un observateur à la classe publisher.

Critique

Ce système a beaucoup d'avantages car il respecte le principe ouvert-fermé, il sépare les préoccupations et permet l'abonnement pendant l'exécution. Cependant, il rends aussi le code plus complexe et plus lent (car si un des observateurs est lent, il ralenti toute l'exécution et c'est généralement assez couteux de le paralléliser). Il ne faut pas l'utiliser tout de suite ou tout le temps. Il faut seulement le faire lorsque c'est vraiment nécessaire (par exemple ne pas le faire si il n'y qu'un observateur).

Revision #2
Created 19 September 2023 18:08:51 by SnowCode
Updated 3 October 2023 21:07:06 by SnowCode