

Patrons de conception

Réutiliser ? Oui. Mais quoi ?

Aurélien Bénel

*Systemes d'information, management des connaissances
et communication*

Bureau T107, aurelien.benel@utt.fr

Qu'est-ce qu'un patron ?

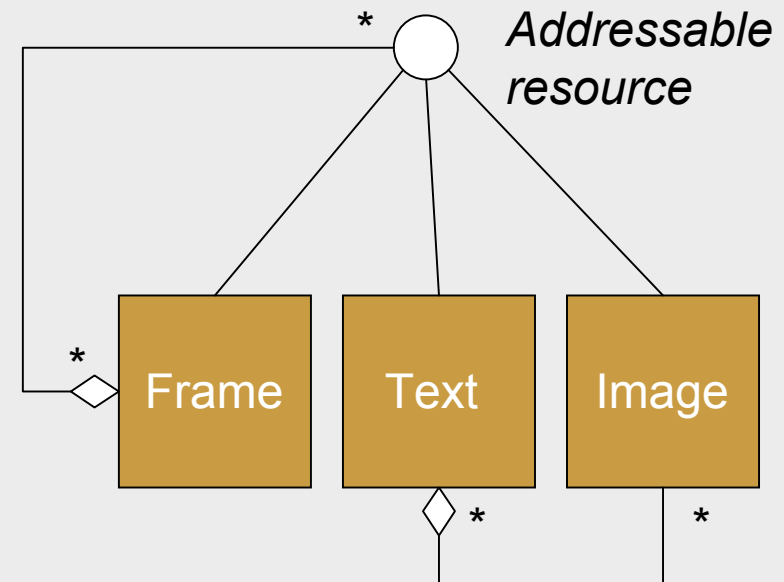
2

« Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. »

(Alexander C., A pattern language: Towns/Buildings/Construction, 1977)

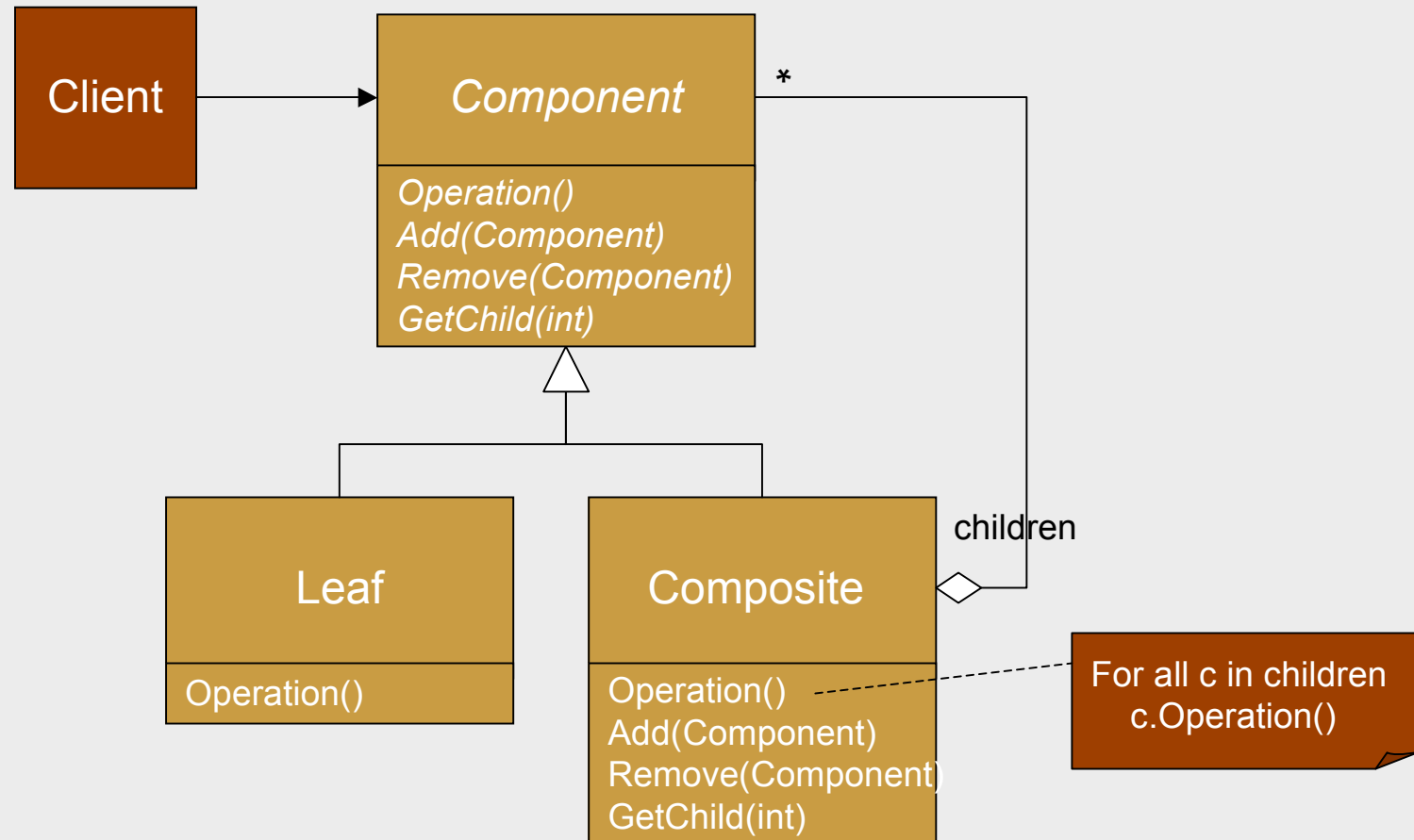
- Nom
- Problème
- Exemple de solution
- Solution générique

- Traiter de la même manière des objets individuels et des compositions d'objets.



Composite

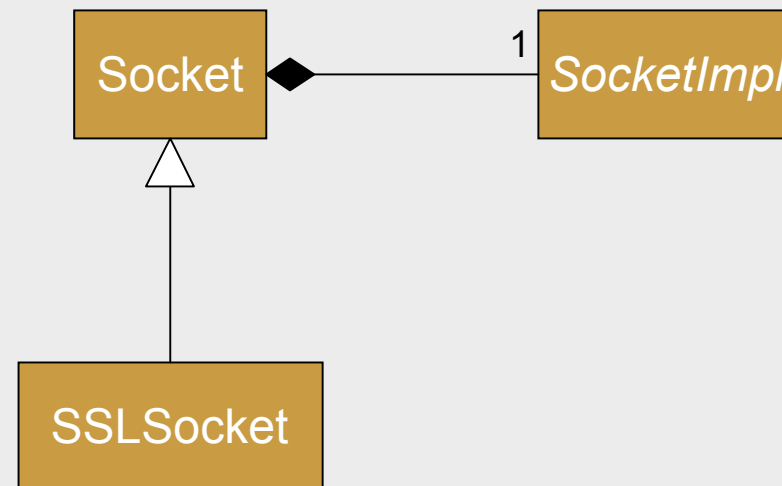
4



Bridge (ou *handle/body*)

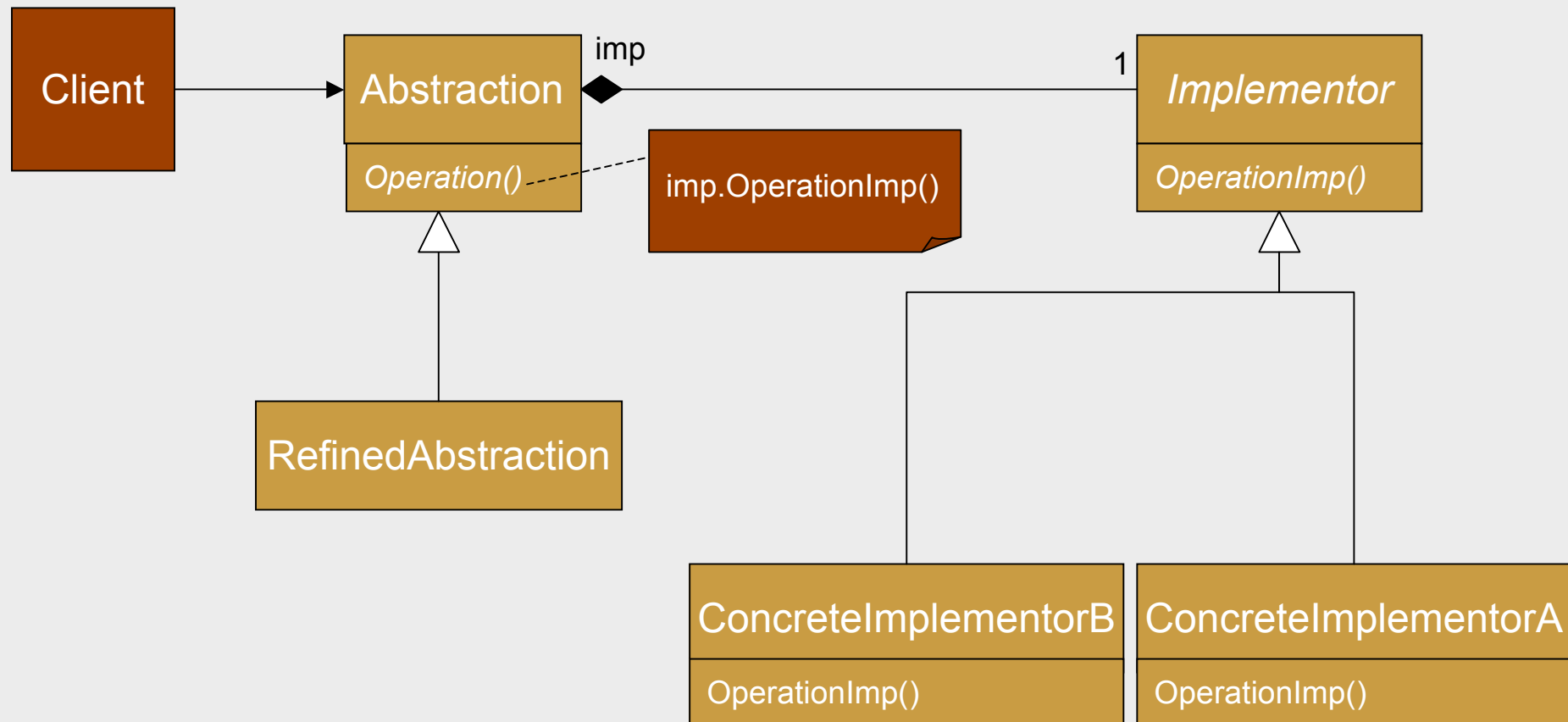
5

- Découpler une abstraction de son **implémentation** de sorte que les deux puissent évoluer (y compris être spécialisés) indépendamment.

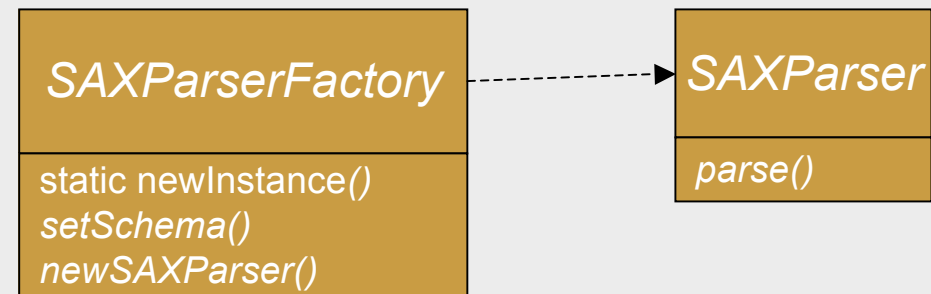


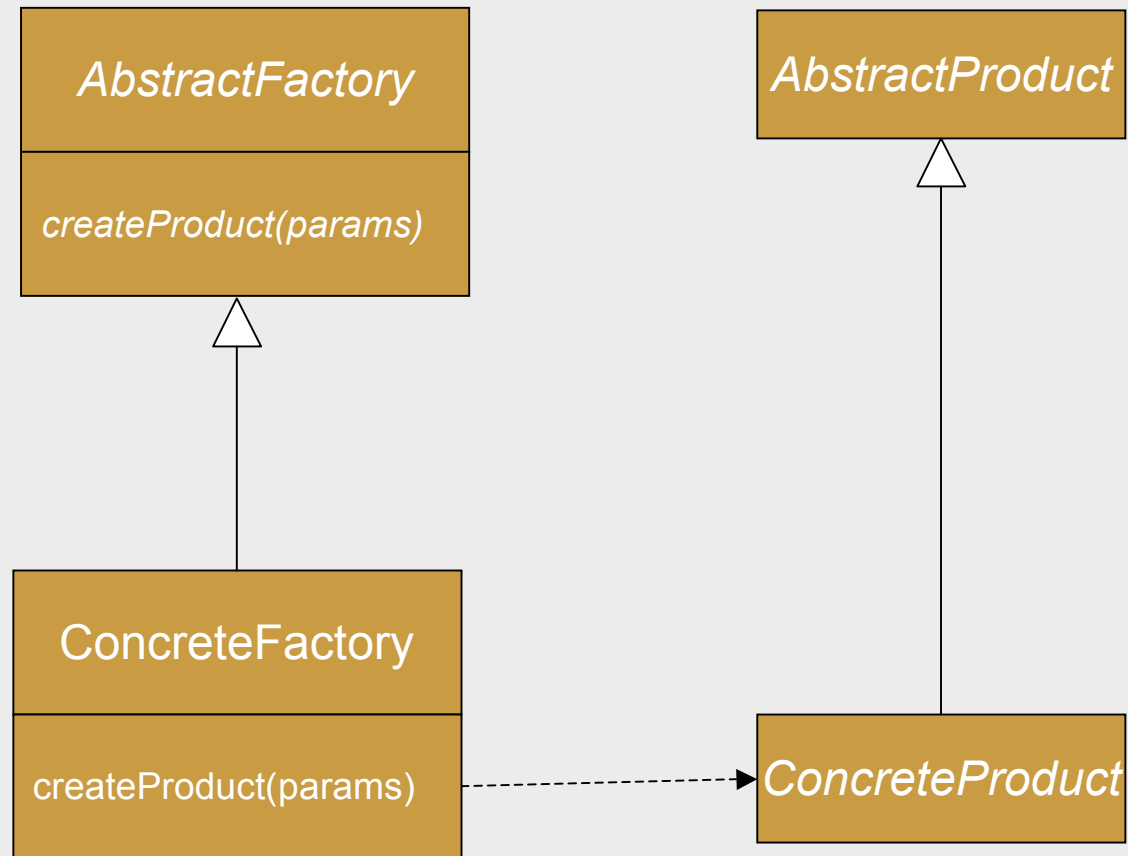
Bridge (ou *handle/body*)

6

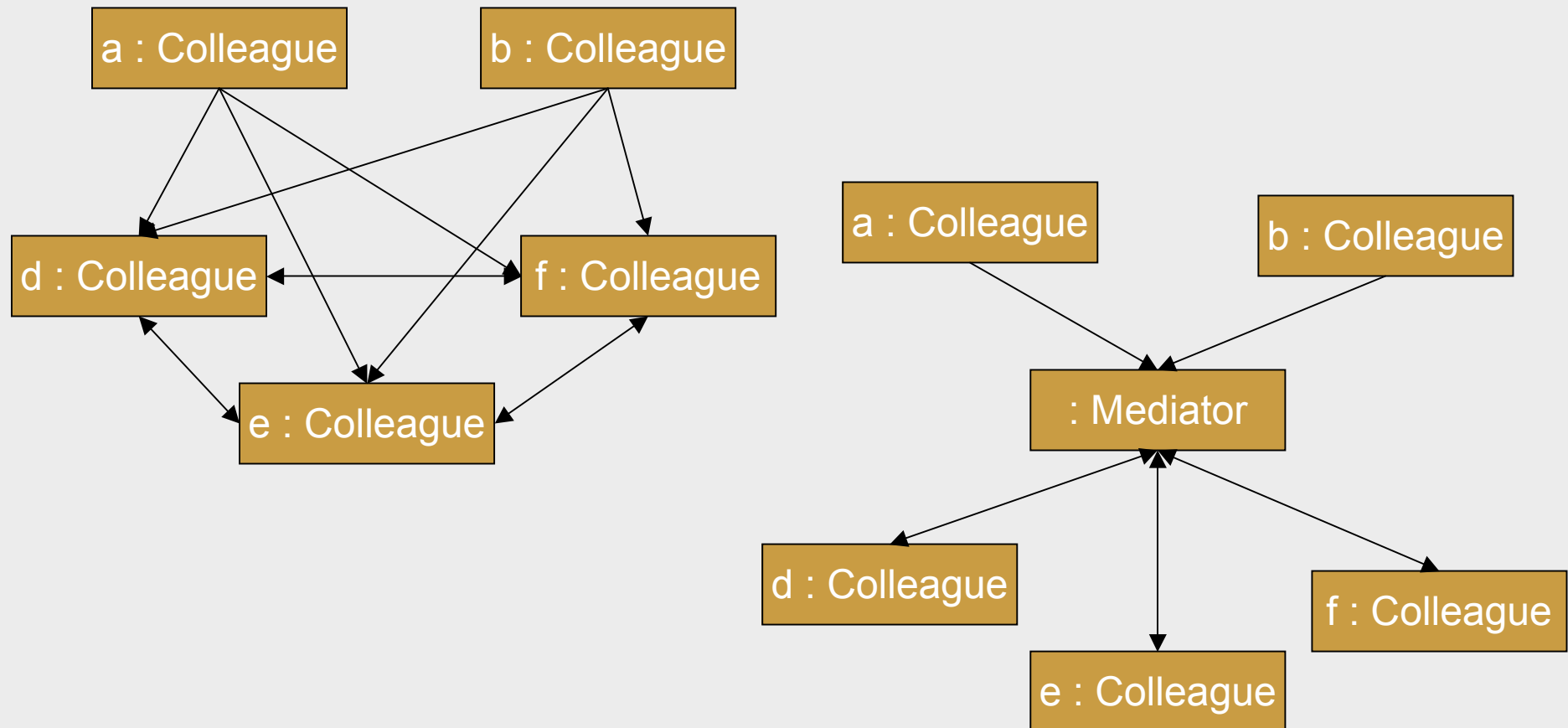


- Laisser des sous-classes décider de la classe à instancier.

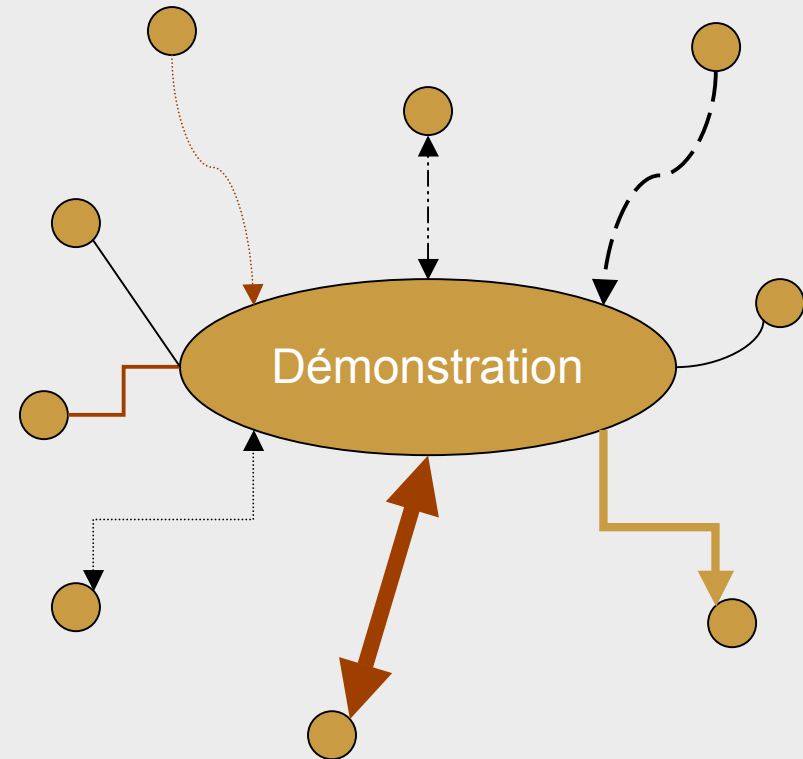


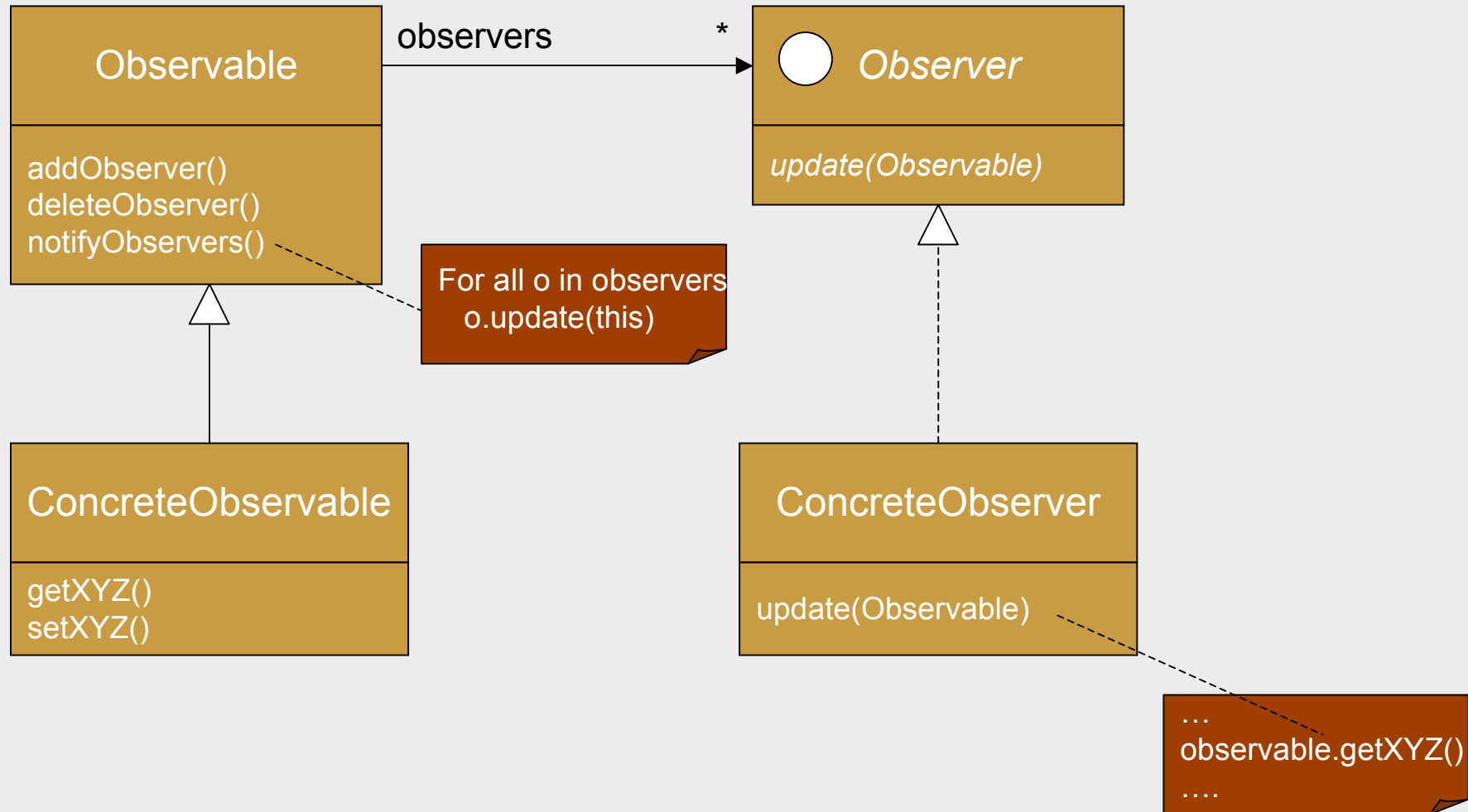


- Réduire le couplage entre objets « collègues ».



- Mettre à jour automatiquement tous les objets qui dépendent d'un objet quand celui-ci est modifié.





- Une aide à la conception objet
- Des mots pour le dire
- Un langage partagé
- Des orientations pour le « refactoring »

- Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Boston : Addison-Wesley, 1994. 395 p.
- Sun Microsystems Inc., *Java 2 Platform Standard Edition 5.0 API Specification*, 2004. Disponible sur : <<http://java.sun.com/j2se/1.5.0/docs/api/>>