Le patron État

[Gamma 95]

État

- Un exemple d’utilisation
- La définition du patron

Un exemple

Implémentation de TCP/IP

TCP/IP
status : String
open() *close()* *send()* *acknowledge()* *synchronize()*

```
TCP/IP::send(s : Stream) {
    if state = 'open'
        {
        (...)
        }

    if state = 'closed'
        {
        (...)
        }

    if state = 'idle'
        {
        (...)
        }
}
```

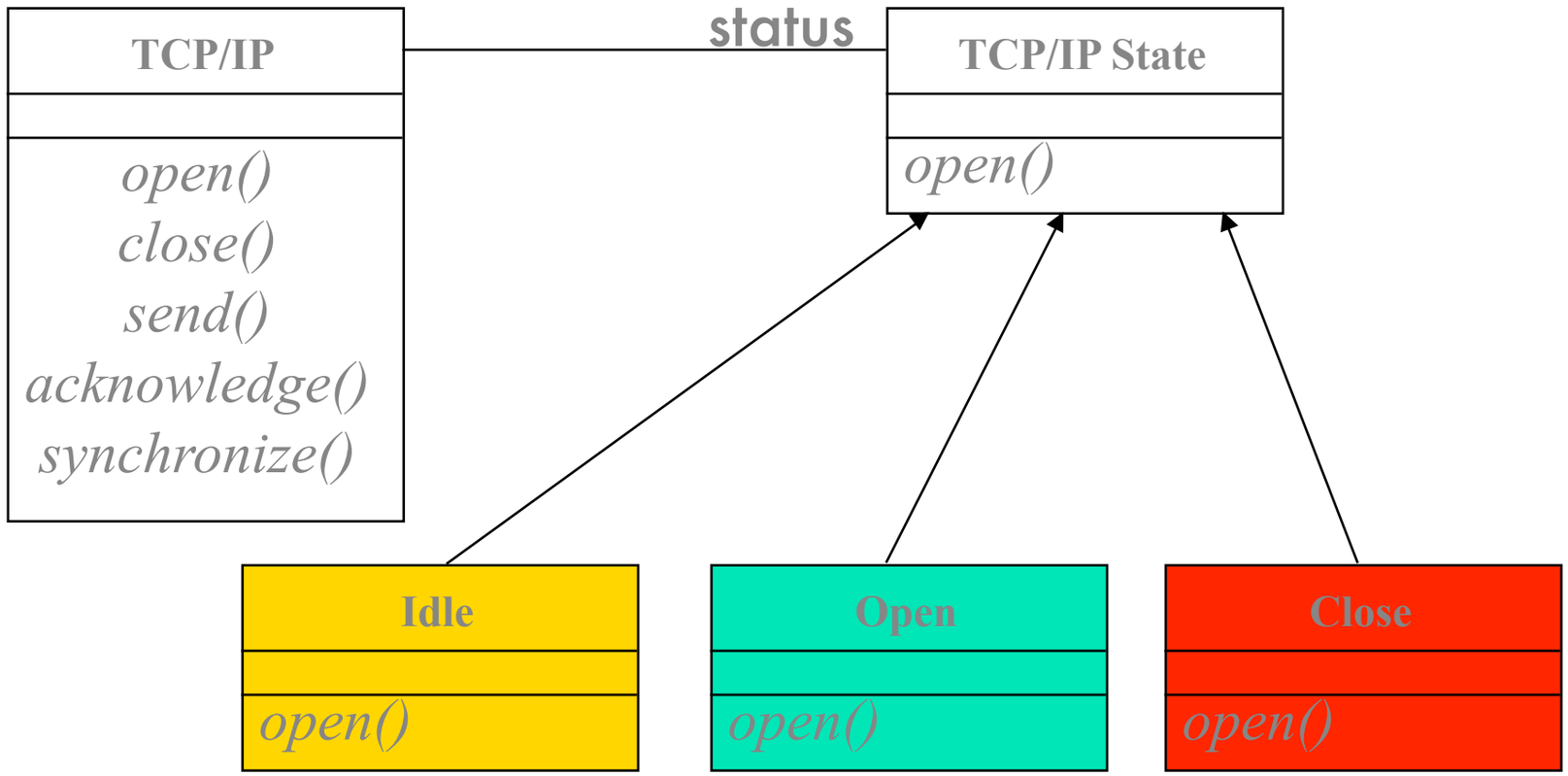
Problème

- Comment éviter que l'état de la connexion soit vérifié à chaque fois qu'un paquet est envoyé?

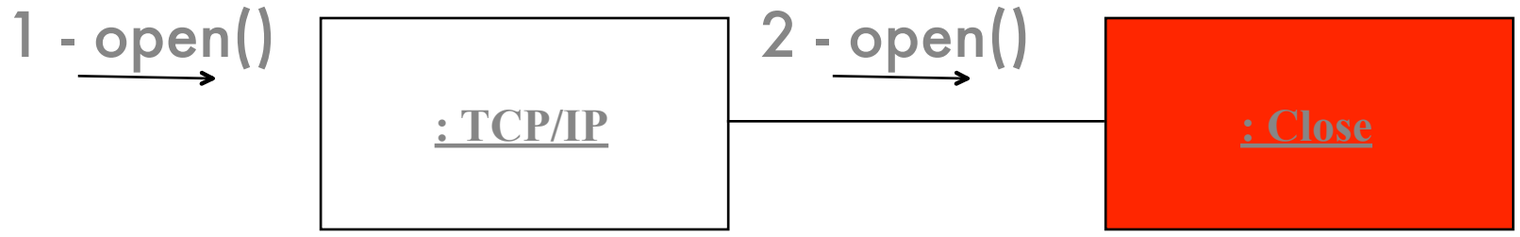
Solution

- Isoler les comportements dépendants des différents états de connexion dans des classes différentes.

En d'autres termes:



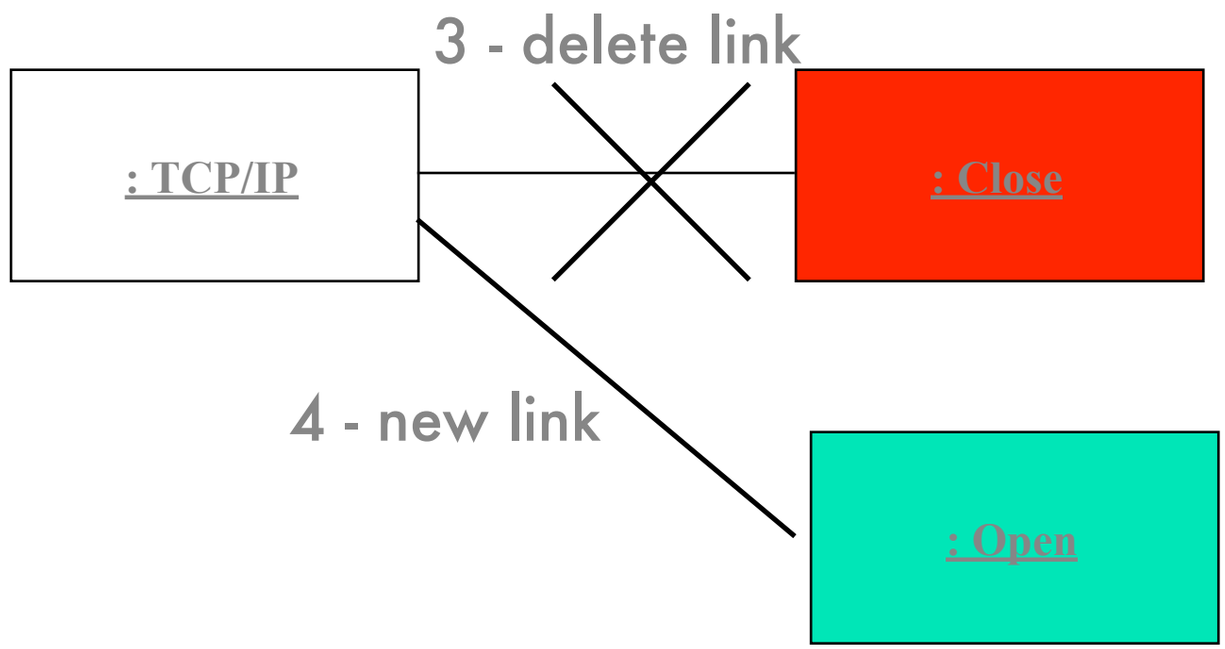
Exemple de connexion



TCP/IP::open() {

this.status.open();
}

Exemple de connexion



Conséquences

- Chaque instance de la classe “TCP/IP” est associée à une instance d’une sous-classe de “TCP/IP State”.
- La vérification de l’état n’est plus nécessaire.

Observations

- Cette solution est utilisée dans la plupart des implémentations du protocole TCP/IP sur Unix.
- Une solution similaire est utilisée dans plusieurs éditeurs graphiques (comportement d'un outil selon le type de figure sélectionnée).

Le patron État

Nom

- State (État).
- Alias: Objects for States, Enveloppe-Letter.

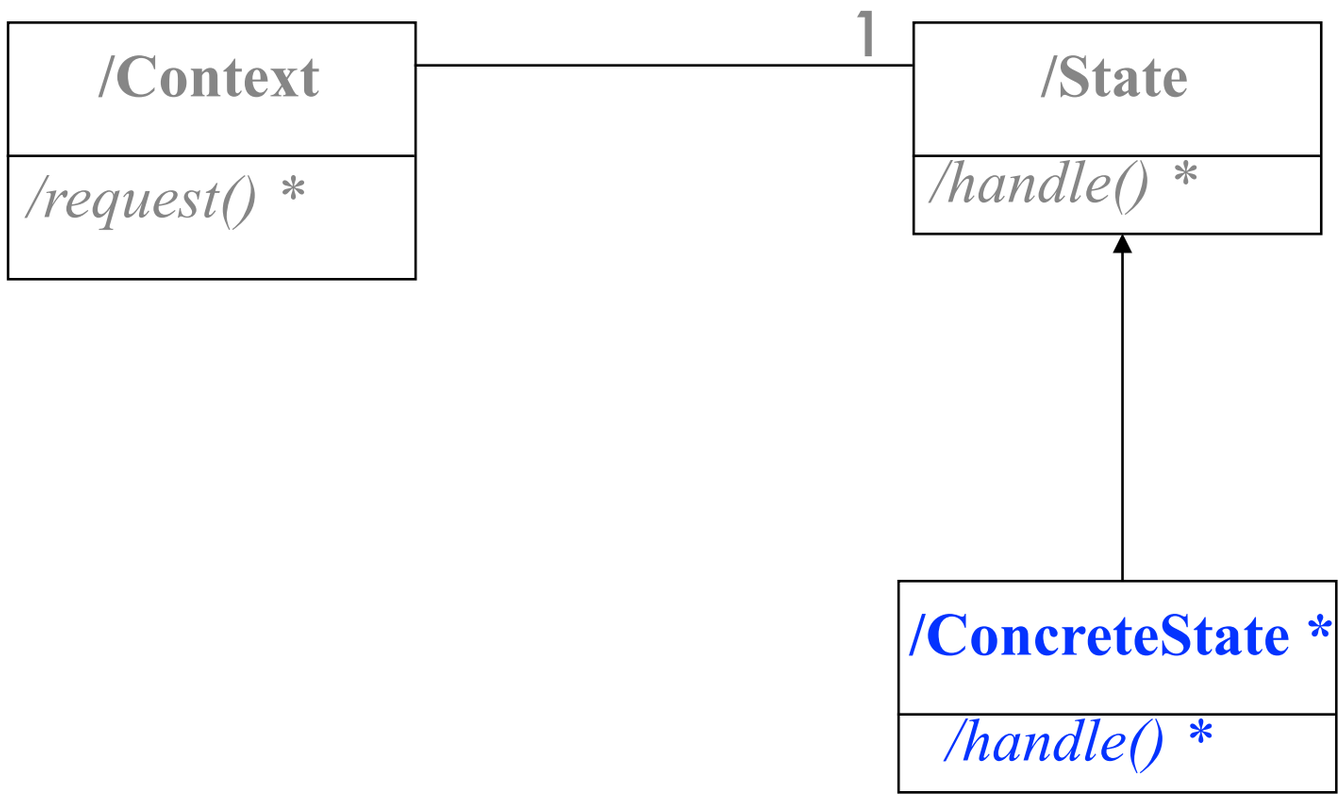
Problème

- Comment traiter un objet dont le comportement est fortement dépendant de son état, sans vérifier, à chaque appel d'une opération, l'état actuel?

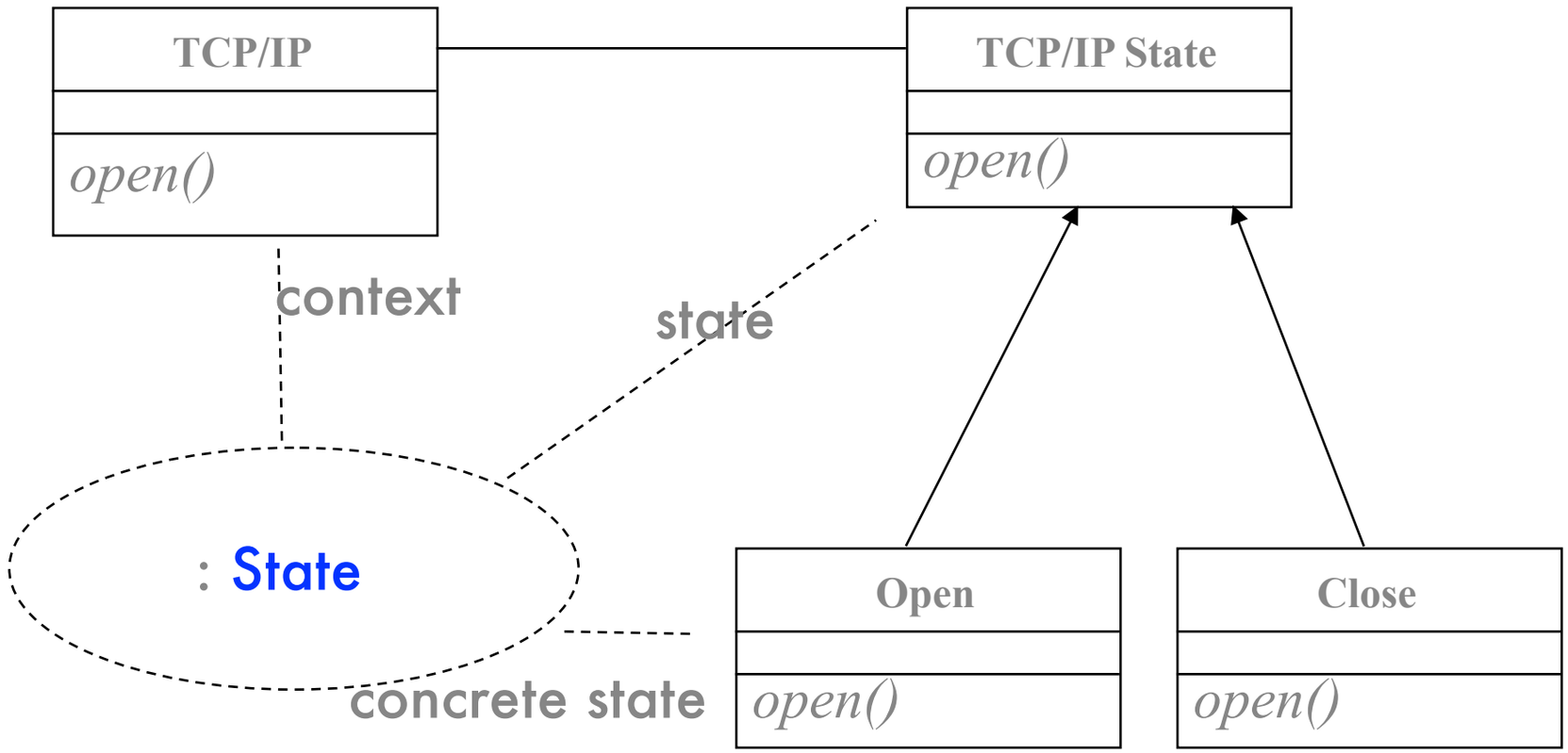
Contexte

- Le comportement d'un objet dépend de son état.
- Différentes opérations contiennent des opérateurs conditionnels sur l'état de l'objet.

Solution (1/2)



Solution (2/2)


TCP/IP
open()
TCP/IP State
open()
context
state
: State
concrete state
Open
open()
Close
open()

Conséquences

- Le comportement spécifique à un état est isolé.
- Les transitions d'état sont explicites.
- Les objets-états peuvent être partagés.

Compromis d'implémentation

- Qui définit les transitions d'état?
- Création et destruction des états.
- Utilisation de l'héritage dynamique.