

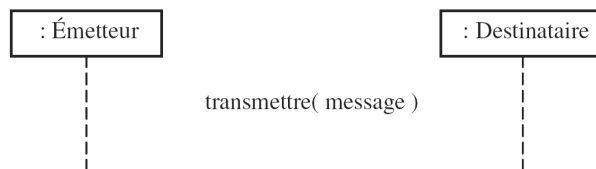
TD4 : Diagramme de séquence UML

Valentin Montmirail
valentin.montmirail@unice.fr

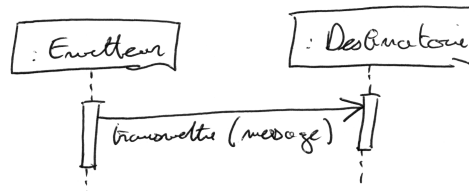
License 3 INFO
Université Nice Sophia-Antipolis

1 Types de messages

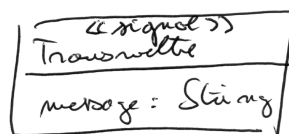
Question : Quand un courrier électronique est envoyé par l'émetteur, celui-ci ne veut pas attendre que destinataire l'ait reçu et il n'y a pas d'intermédiaire. Peut-on utiliser un message synchrone ? Complétez la figure ci dessous par des flèches représentant des messages.



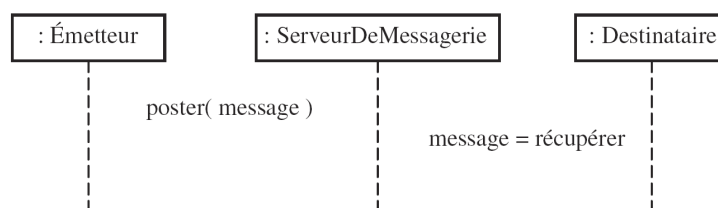
Un message asynchrone est indiqué ici.



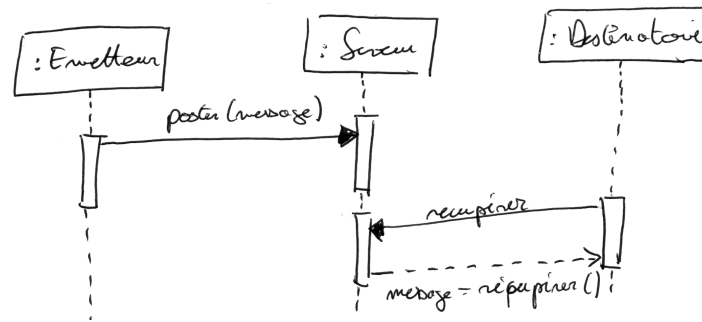
Question : Est-ce que transmettre est une opération ou un signal ? Dans tous les cas, donnez des éléments d'un diagramme de classe cohérent avec le diagramme de séquence.



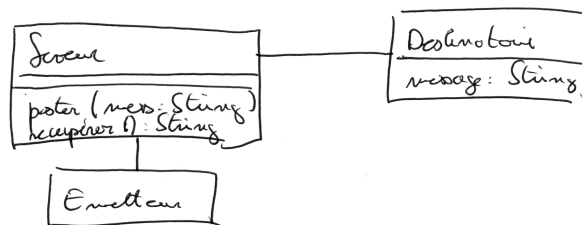
Question : Un serveur de messagerie sert d'intermédiaire entre l'émetteur et le récepteur d'un email. Le serveur est toujours en fonction. Est-ce qu'on peut utiliser des messages synchrones pour l'envoi et la récupération de emails ? Complétez la figure ci dessous par des flèches représentant des messages.



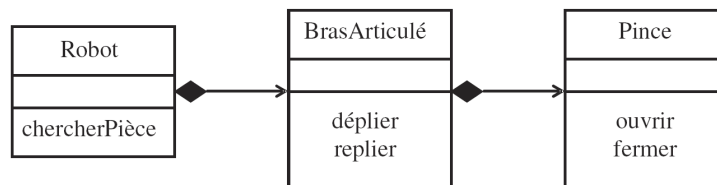
Un message synchrone est possible ici et c'est donc préférable : si on a le choix, il vaut mieux utiliser des messages synchrones, qui s'implémentent facilement par des opérations.



Question : Est-ce que poster est une opération ou un signal ? Dans tous les cas, proposez un diagramme de classe cohérent avec le diagramme de séquence.



2 Modélisation de la dynamique de l'interaction entre classes



Le diagramme de classes ci-dessus décrit des robots composés de bras articulés, eux-mêmes composés de pinces. L'implémentation des classes en question est la suivante :

```

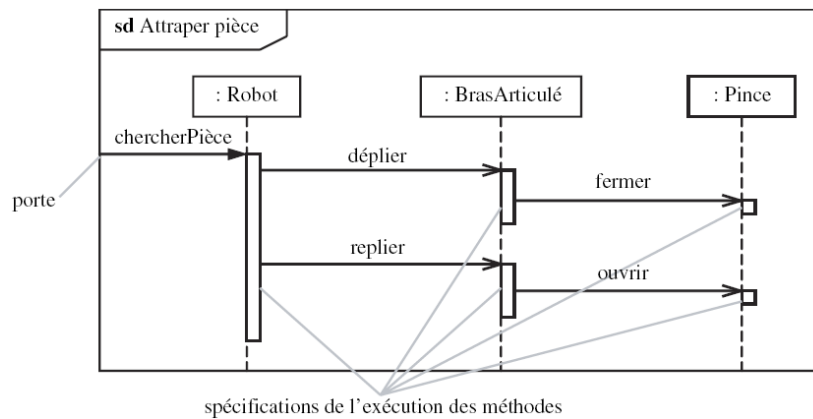
class Robot{
privée :
    BrasArticulé brasArticulé ;
publique :
    void chercherPièce() {
        brasArticulé.déplier() ;
        brasArticulé.replier() ;
    }
}
class BrasArticulé {
privée :
    Pince pince ;
publique :
    void déplier() {
        ...
        pince.fermer() ;
    }
    void replier() {
        ...
    }
}
  
```

```

        pince.ouvrir() ;
    }
}
class Pince {
privée :
    ...
publique :
    void fermer() { ... }
    void ouvrir() { ... }
}
Début programme principal
    Robot robot ;
    robot.chercherPièce() ;
Fin programme principal

```

Question : Donnez un diagramme de séquence représentant l'échange de messages résultant de l'exécution du programme principal.¹



NB : ce sont des messages synchrones qu'il faut utiliser.

3 Syntaxe des messages

Question : Expliquez la syntaxe des messages suivants, extraits d'un diagramme de séquence :

- f
- f(0)
- f(x)
- f(x=0)
- f(y=x)
- f(.)
- f(x,y)
- *
- y=f
- y=f(0)
- y=f(x=0)
- y=f(x):0

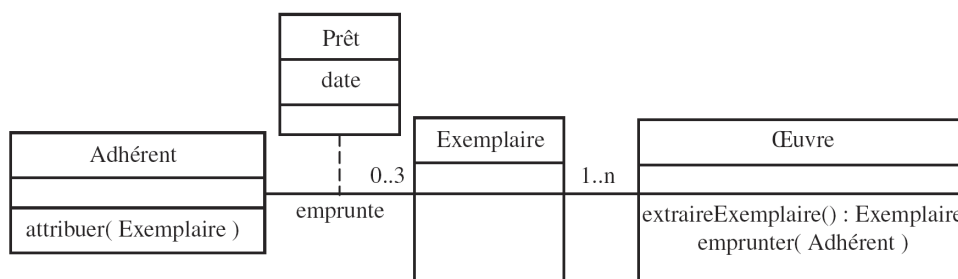
- *f est un message sans argument.*
- *f(0) est un message qui reçoit en argument la valeur 0.*
- *f(x) est un message qui reçoit la valeur de x en argument.*
- *f(x = 0) est un message qui reçoit un argument x ayant pour valeur 0.*
- *f(y = x) est un message ayant un argument y qui prend la valeur de x.*
- *f(-) est un message avec un argument non défini.*

1. On pourra faire partir chercherPièce() du bord du cadre de la séquence

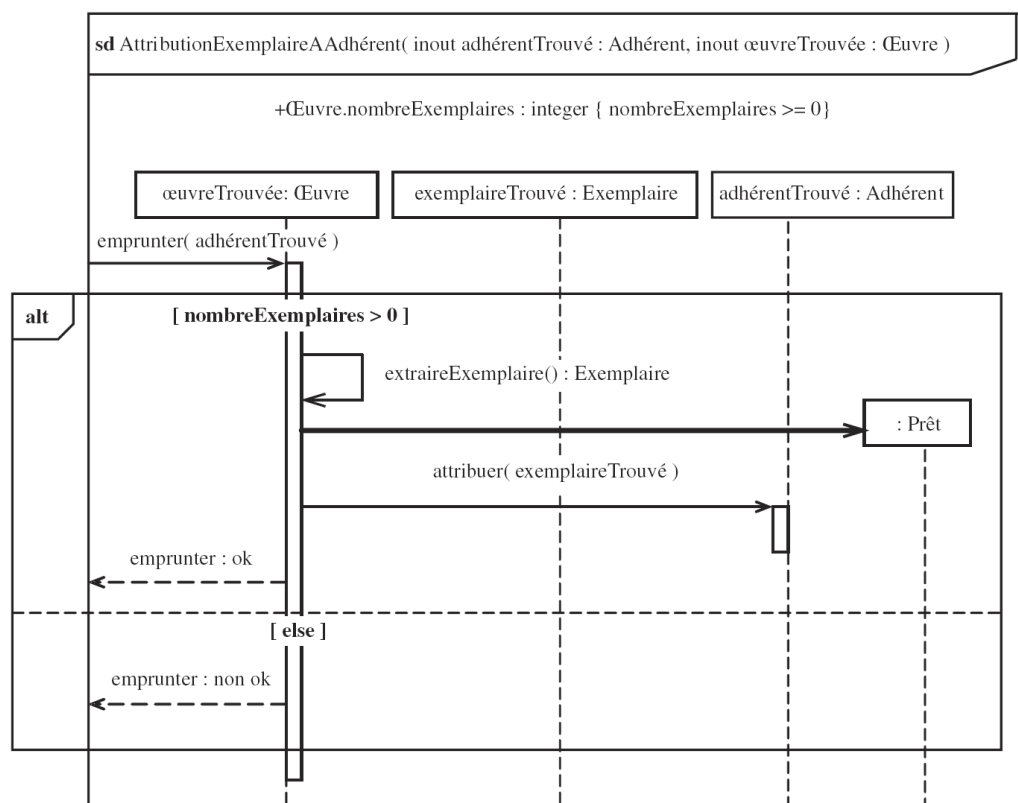
- $f(x, y)$ est un message qui reçoit en arguments les valeurs de x et de y .
- $*$ est un message de type quelconque.
- $y = f$ est un message de réponse à un message f ; la valeur de retour est affectée à y .
- $y = f(0)$ est un message de réponse à un message $f(0)$; la valeur de retour est affectée à y .
- $y = f(x = 0)$ est un message de réponse à un message $f(x = 0)$; la valeur de retour est affectée à y .
- $y = f(x) : 0$ est un message de réponse à un message $f(x)$; la valeur de retour 0 est affectée à y .

4 Modélisation d'une interaction interne

Le diagramme de classes présenté ci dessous modélise la structure interne de la bibliothèque.



Un acteur adhérent peut emprunter un exemplaire d'une oeuvre donnée. L'emprunt se fait de la façon suivante : la méthode emprunter est appelée avec un objet de classe Adhérent donné en argument ; s'il reste des exemplaires dans la bibliothèque, l'un des exemplaires associés à l'oeuvre est extrait via la méthode extraireExemplaire, une instance de la classe Prêt est créée, puis l'exemplaire extrait de la bibliothèque est attribué à l'adhérent grâce à l'opération attribuer. S'il restait un exemplaire, l'oeuvre retourne « OK » et dans le cas contraire, elle retourne « PasOK ».



5 Documentation d'un cas d'utilisation

La rubrique « enchaînement nominal » du cas d'utilisation « retrait d'espèces » contient les éléments suivants :

1. Le guichetier saisit le numéro de compte du client ;
2. L'application valide le compte auprès du système central ;
3. Le guichetier demande un retrait de 100 euros ;
4. Le système « guichet » interroge le système central pour s'assurer que le compte est suffisamment approvisionné ;
5. Le système central effectue le débit du compte ;
6. En retour, le système notifie au guichetier qu'il peut délivrer le montant demandé.

Question : Donner le diagramme de séquences associé à cette description textuelle.

