UML2

Diagrammes de séquences

Réf. ”Introduction à UML2” Miles, Hamilton. Ed. O'Reilly. 2006

UML2 : Diagrammes de séquences

Modéliser le ”comment”

→ diagrammes d'interactions (à l'exécution)

Diagrammes de séquences :

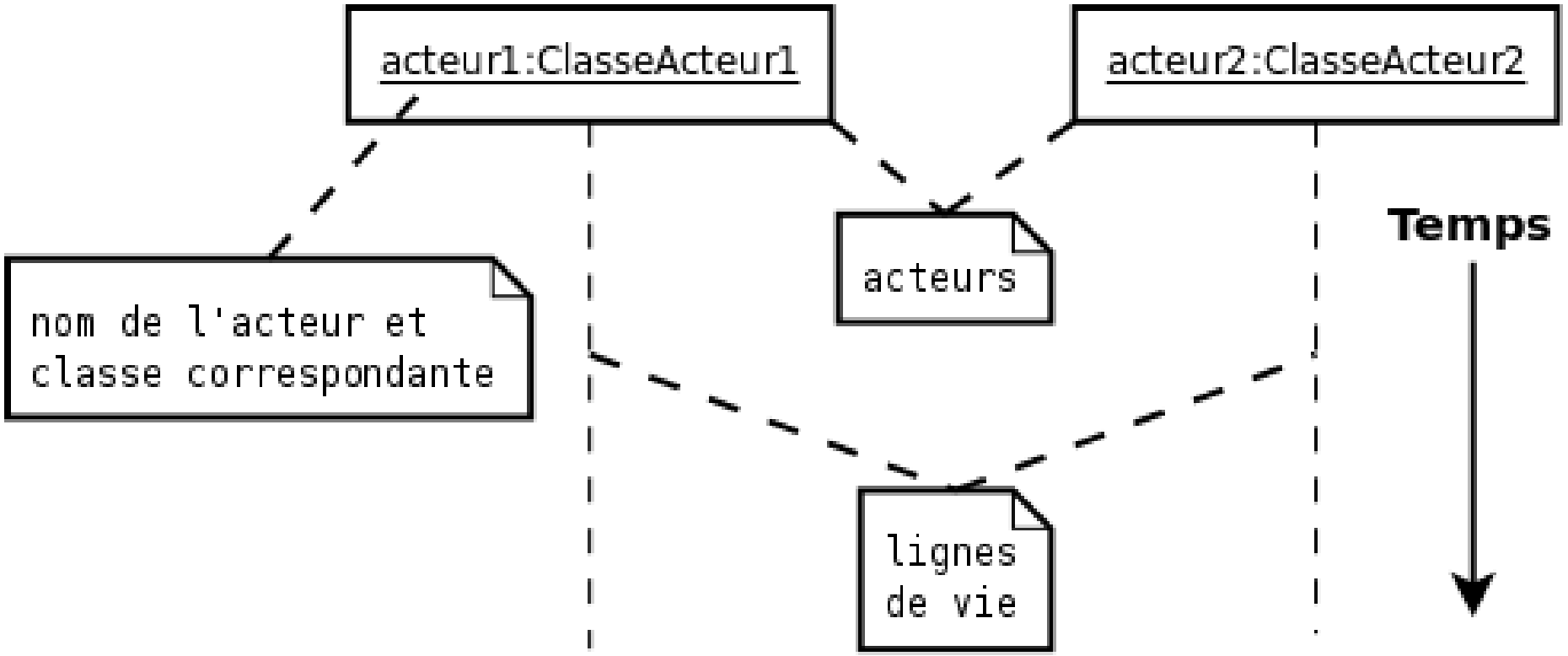
- Ordre (séquence) des interactions
- vue logique du système

Autres diagrammes d'interactions :

- diagrammes de communication
- diagrammes de chronométrage

UML2 : Diagrammes de séquences

Acteurs d'un diagramme de séquences



UML2 : Diagrammes de séquences

Nom d'acteur

nom [sélecteur] : *nom_de_classe* ref *décomposition*

Exemple de nom	Description
admin	Partie du système nommée *admin*, mais à ce stade, aucune classe ne lui a été attribuée.
: SystèmeDeGestionDeContenu	La classe de l'acteur est *SystèmeDeGestionDeContenu*, mais la partie n'a pas encore son propre nom.
admin : Administrateur	Une partie de nom *admin* et de classe *Administrateur*
gestionnairesÉvenements [2] : GestionnaireÉvenements	Partie à laquelle on accède par l'élément 2 d'un tableau et de classe *GestionnaireÉvenements.*
: SystèmeDeGestionDeContenu ref interactionsCMS	La classe de l'acteur est *SystèmeDeGestionDeContenu*, et il existe un autre diagramme d'interactions nommé *interactionsCMS* qui connaît le fonctionnement interne de cet acteur.

UML2 : Diagrammes de séquences

RMQ :

UML 1 : diagrammes d'interactions → objets

UML2 :

+ général

diagrammes d'interactions → parties du système

acteur : objet ou toute partie du système

UML2 : Diagrammes de séquences

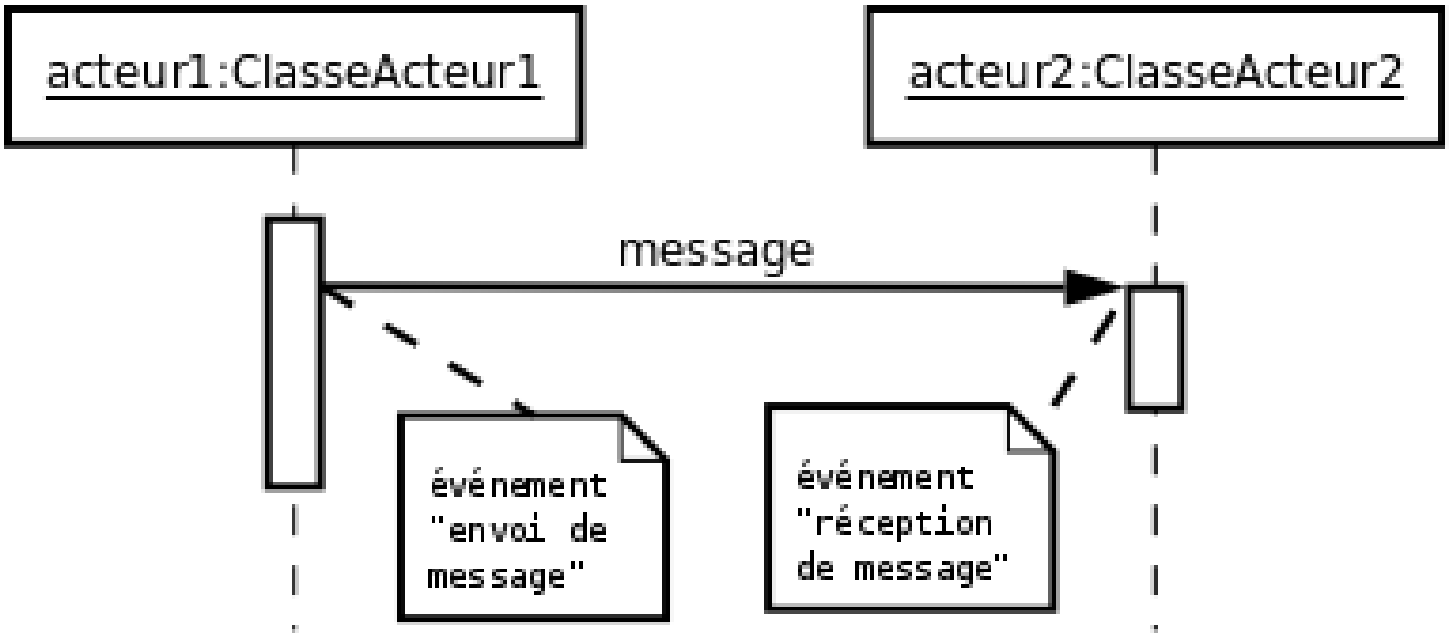
Temps

- s'écoule de haut en bas
- → ordre des interactions
- ne représente pas les durées

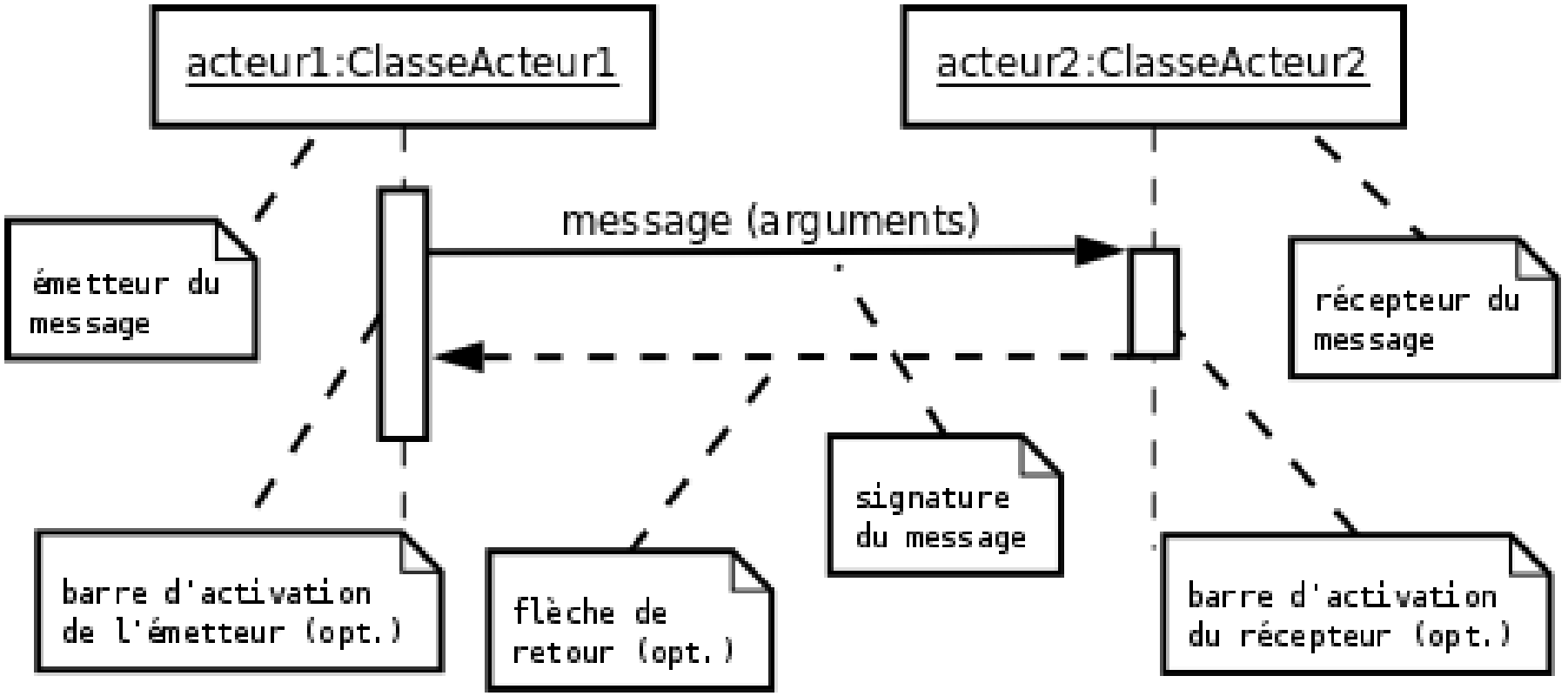
Durées → diagrammes de chronométrage

UML2 : Diagrammes de séquences

Évenemenents, signaux et messages



UML2 : Diagrammes de séquences



UML2 : Diagrammes de séquences

Signature d'un message

attribut = nom_message (arguments) : type de retour

Format d'un argument

nom : classe

Nombre quelconque d'arguments : 0 .. n

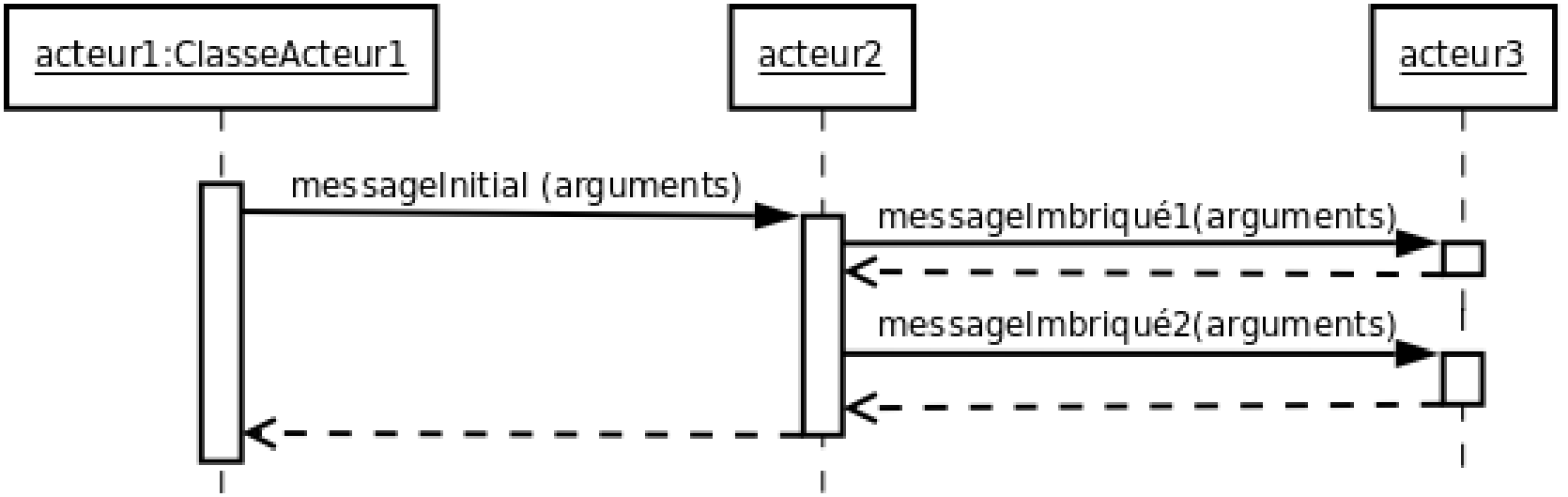
2 ou + : séparés par une virgule

UML2 : Diagrammes de séquences

Exemple de signature	Description
faireQuelqueChose()	Le nom du message est faireQuelqueChose, mais nous n'avons pas plus d'informations à son propos.
faireQuelqueChose(nombre1 : Nombre, nombre2 : Nombre)	Le nom du message est faireQuelqueChose et il prend 2 arguments, nombre1 et nombre2, tous 2 de la classe Nombre.
faireQuelqueChose() : ClasseDeRetour	Le nom du message est faireQuelqueChose, il ne prend aucun argument, et renvoie un objet de la classe ClasseDeRetour.
MaVar = faireQuelqueChose() : ClasseDeRetour	Le nom du message est faireQuelqueChose, il ne prend aucun argument, et renvoie un objet de la classe ClasseDeRetour affecté à l'attribut maVar de l'émetteur du message.

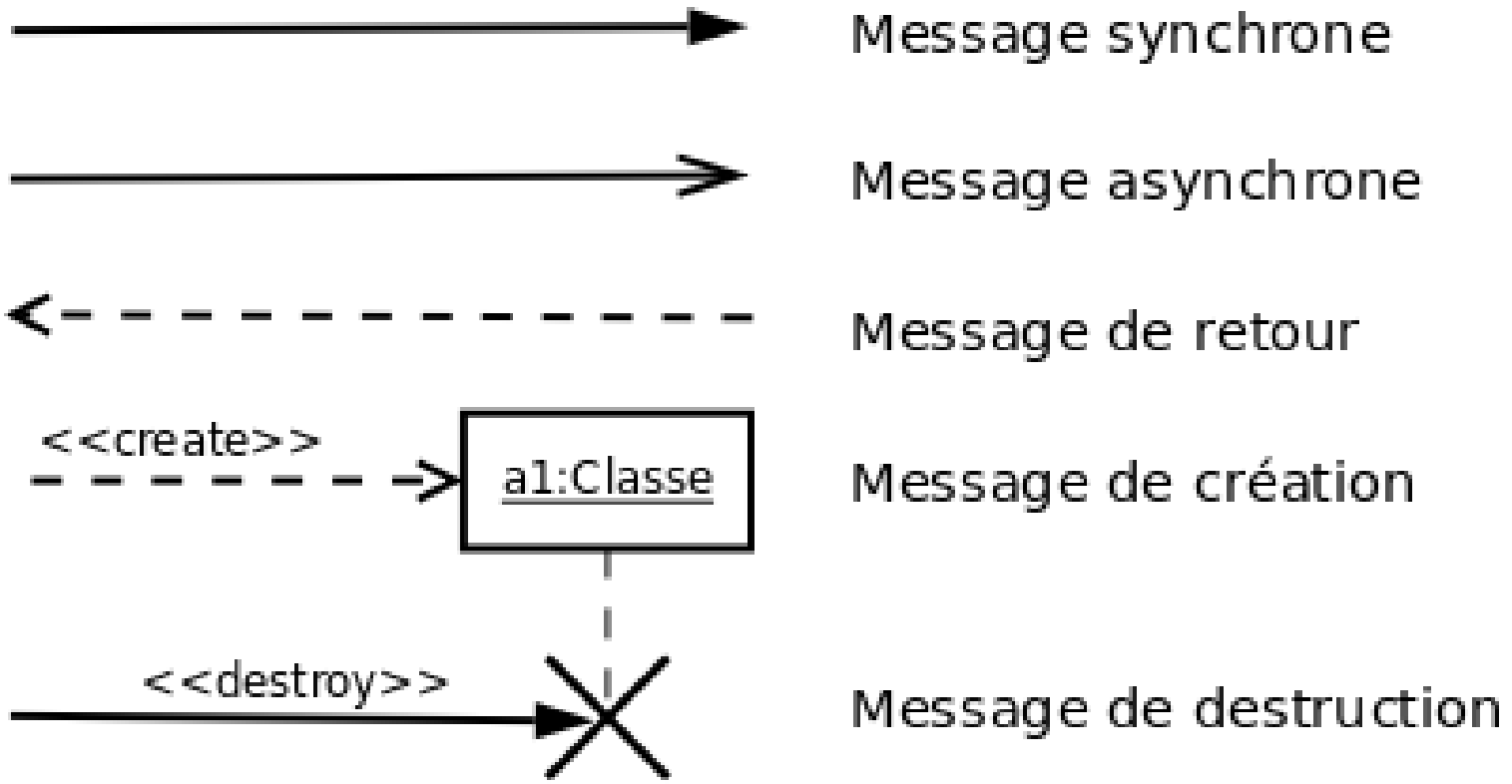
UML2 : Diagrammes de séquences

- Barres d'activité
- Messages imbriqués



UML2 : Diagrammes de séquences

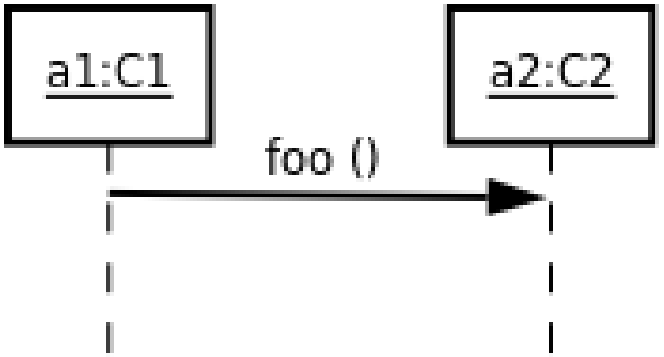
- Flèches de messages



UML2 : Diagrammes de séquences

Messages synchrones

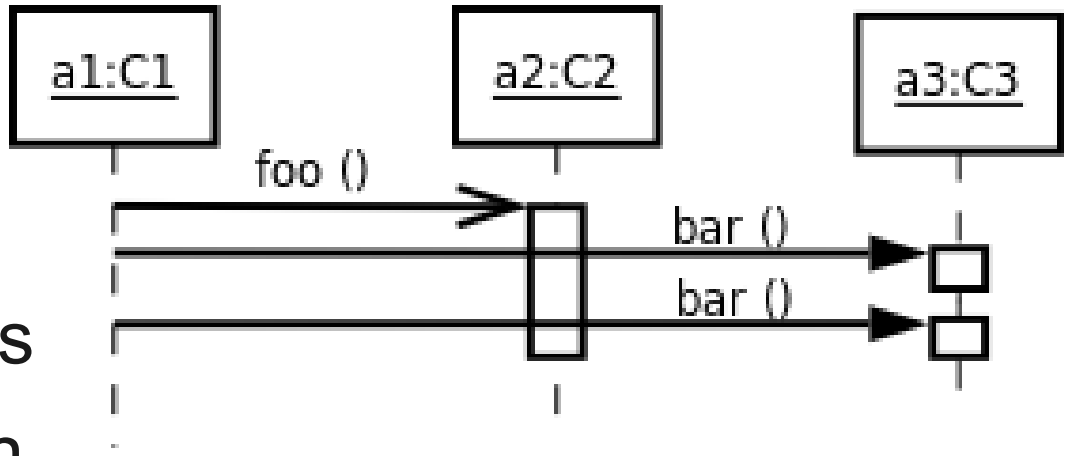
l'émetteur attend
que le récepteur
ait terminé



→ appel de méthode

UML2 : Diagrammes de séquences

Messages asynchrones



L'émetteur n'attend pas

le retour de l'invocation

pour continuer

→ souvent implémentés par des threads

UML2 : Diagrammes de séquences

Message de retour

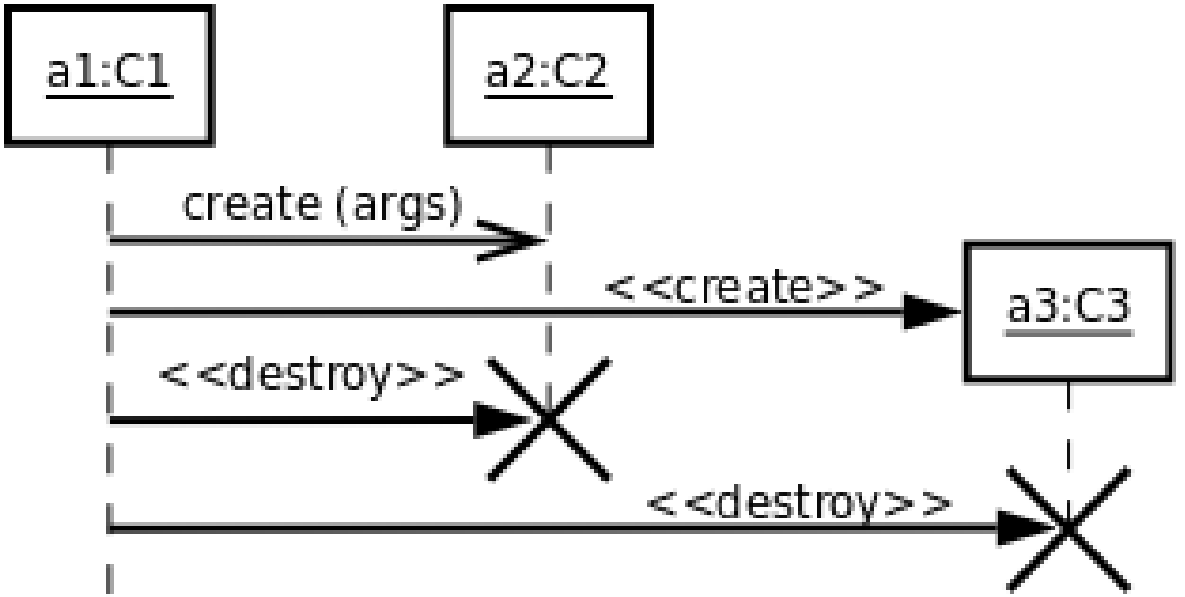
- facultatif
- le flux de contrôle revient à l'émetteur du message

Rmq : un acteur peut s'envoyer un message

- pour montrer une décomposition
- codage : appel de méthode (this)

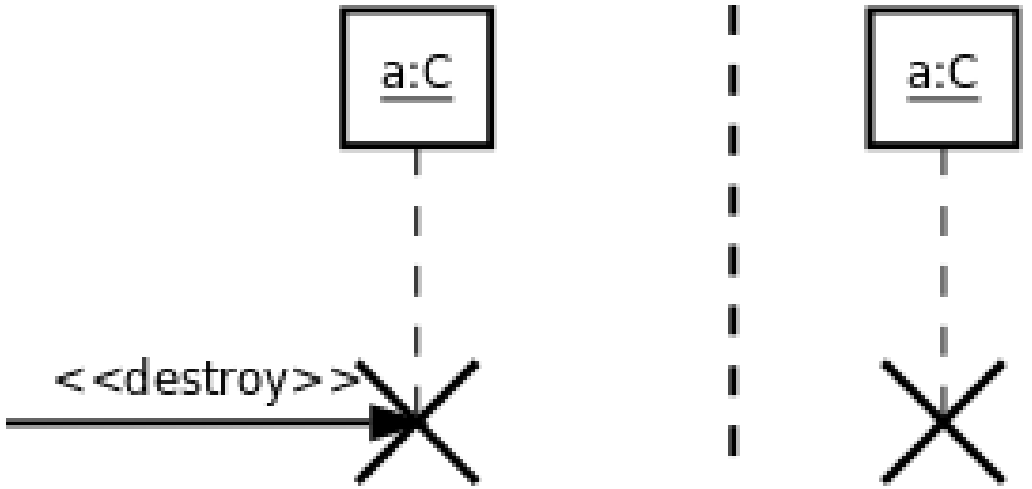
UML2 : Diagrammes de séquences

Messages de création et de destruction



UML2 : Diagrammes de séquences

Destruction explicite / implicite



Ex. Java garbage collector

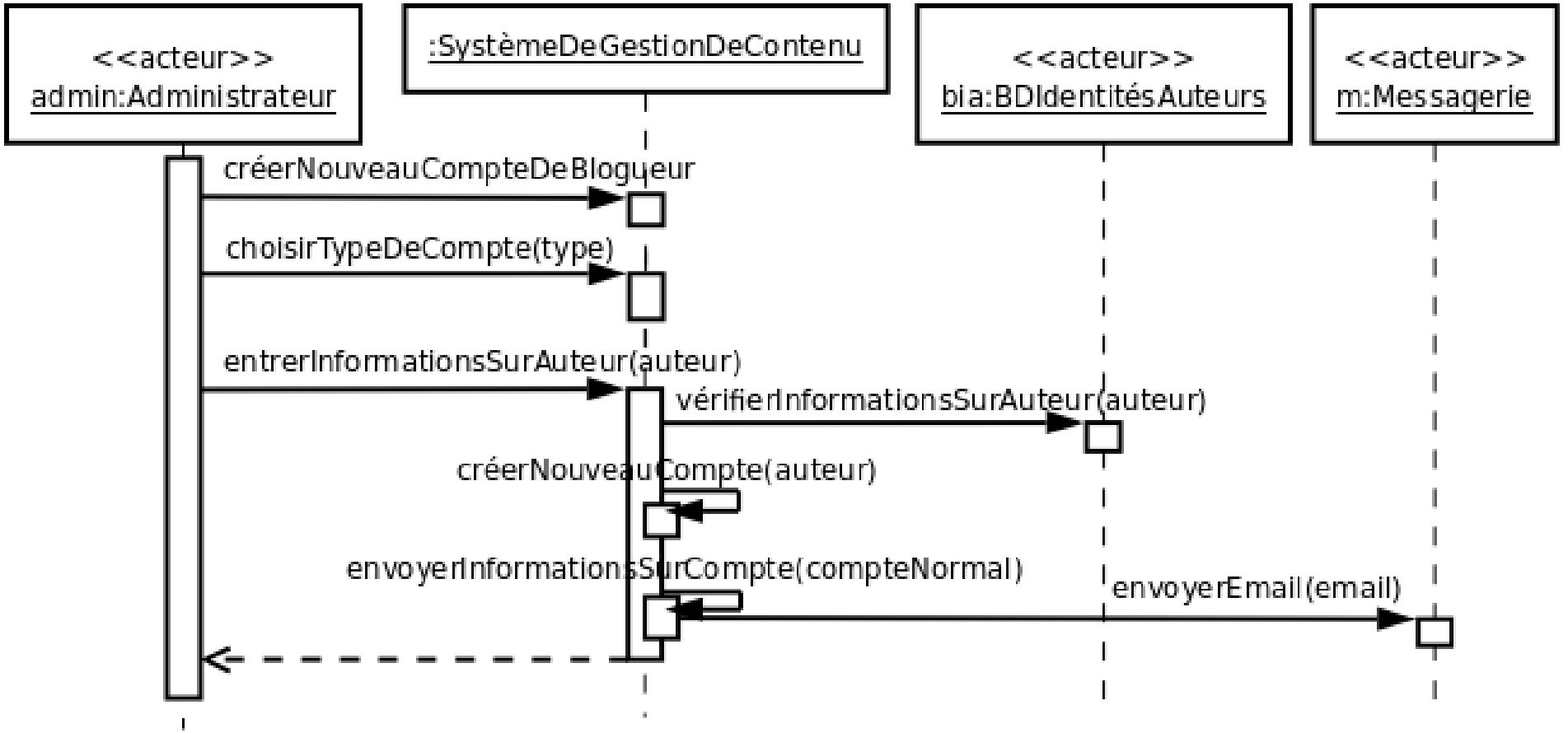
UML2 : Diagrammes de séquences

Donner vie à un cas d'utilisation

Ex. Cas d'utilisation ”Créer un compte de blogueur”

- diagramme de séquence de haut niveau
- point de départ : flux principal (sans extension)
- focalisation sur les acteurs et les messages

UML2 : Diagrammes de séquences



UML2 : Diagrammes de séquences

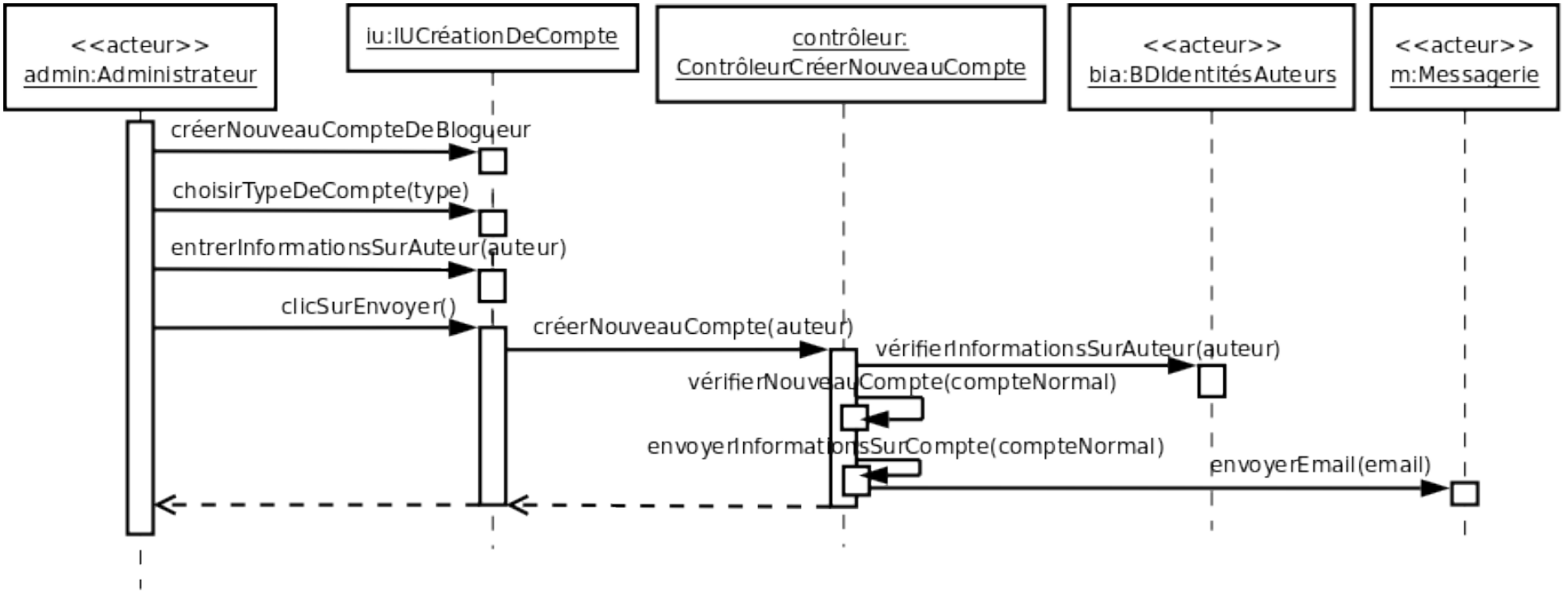
Décomposer une interaction en différents acteurs

- acteurs + système (niveau des U.C.)
- → décomposer l'acteur système :
  - parties ”métier”
  - parties ”IHM”
  - parties ”contrôle”, ...

→ modifications des diagrammes pour trouver :

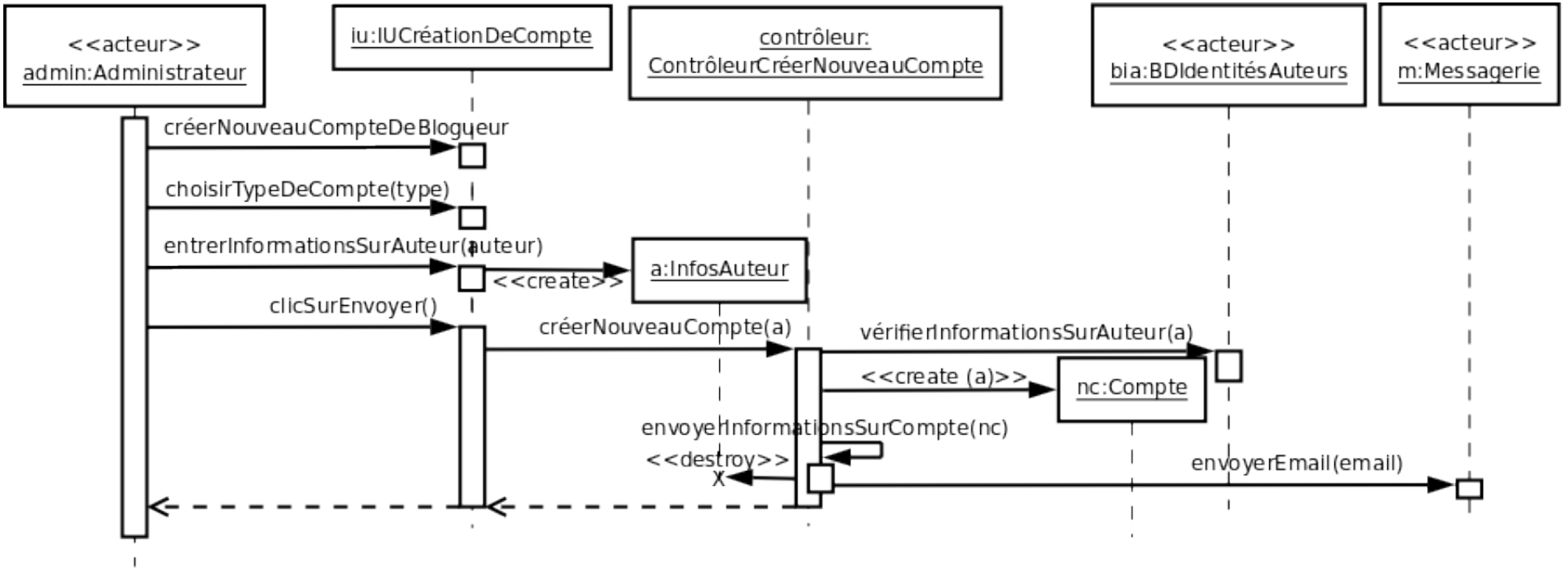
- les bons acteurs
- les bonnes interactions

UML2 : Diagrammes de séquences



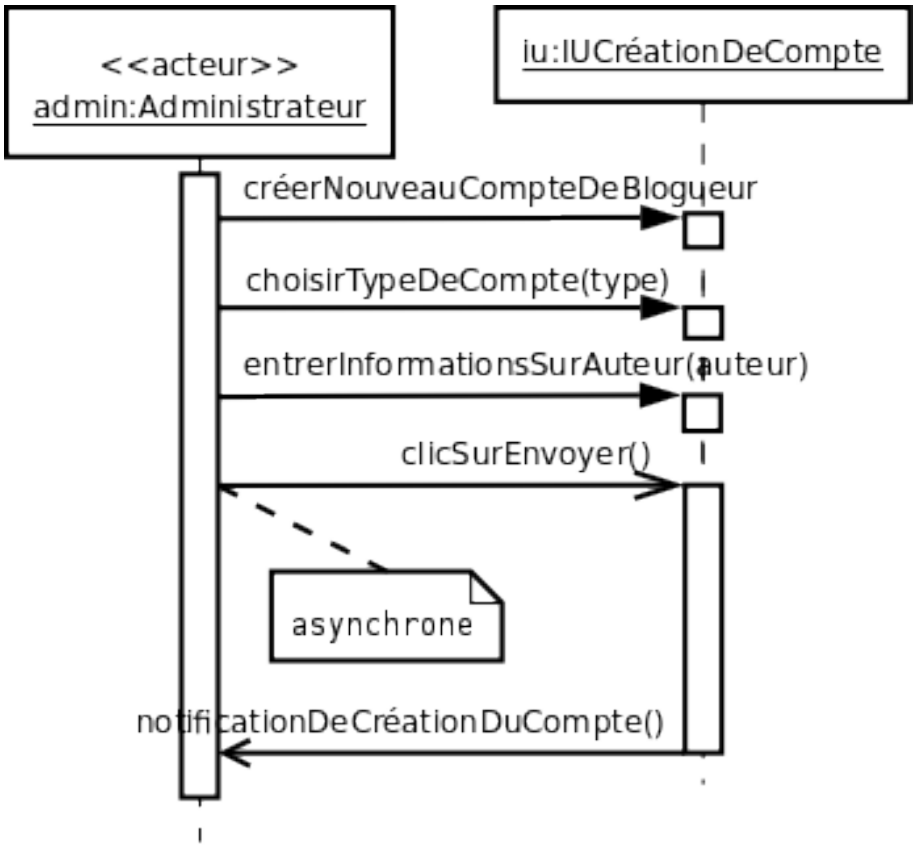
UML2 : Diagrammes de séquences

Faire apparaître la création / destruction d'acteurs



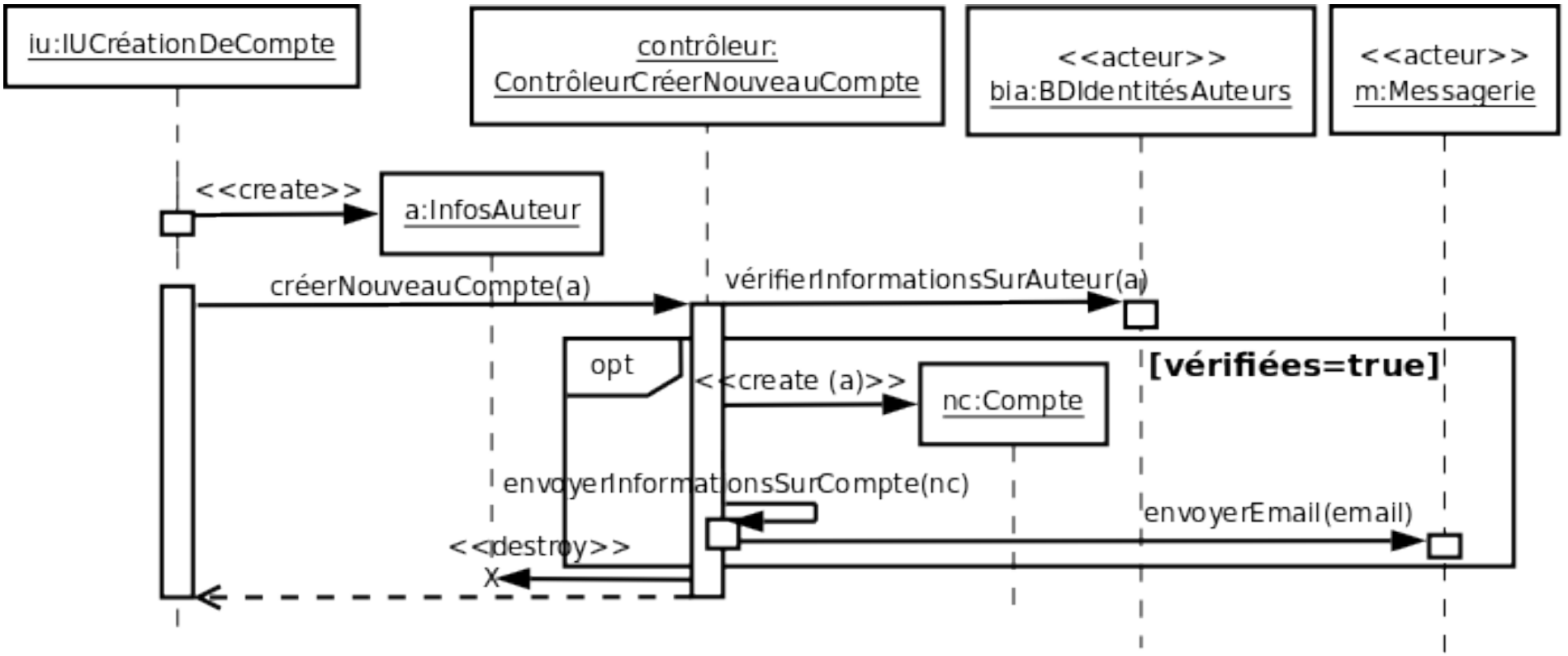
UML2 : Diagrammes de séquences

Faire apparaître les messages asynchrones



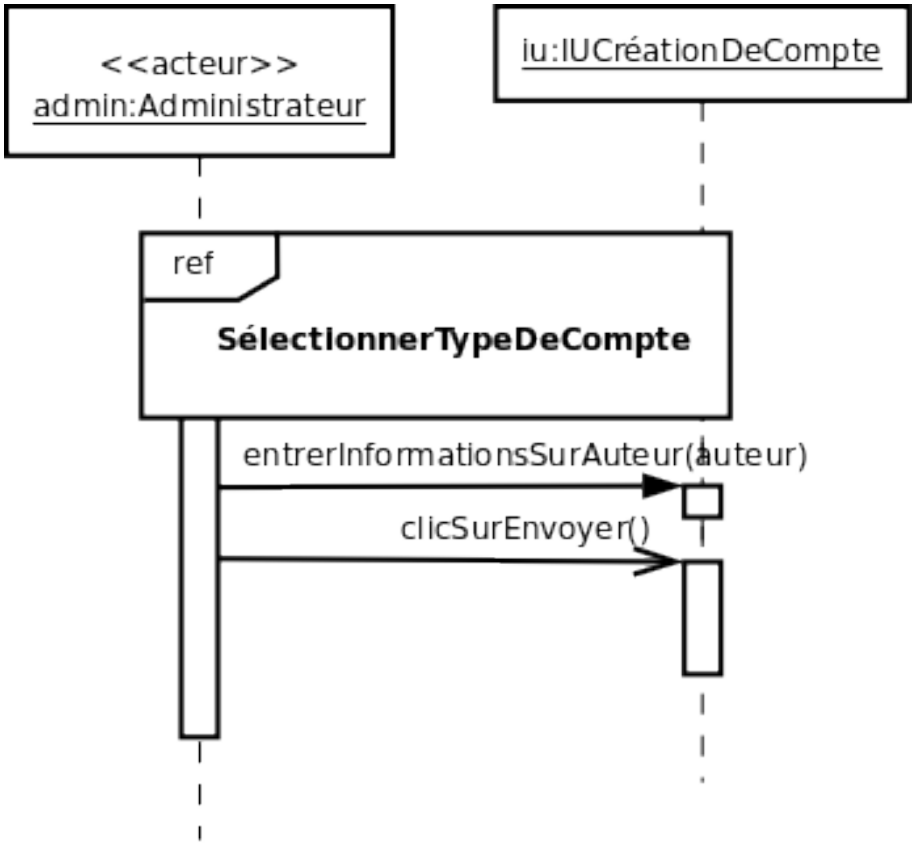
UML2 : Diagrammes de séquences

Gérer les interactions complexes avec des fragments



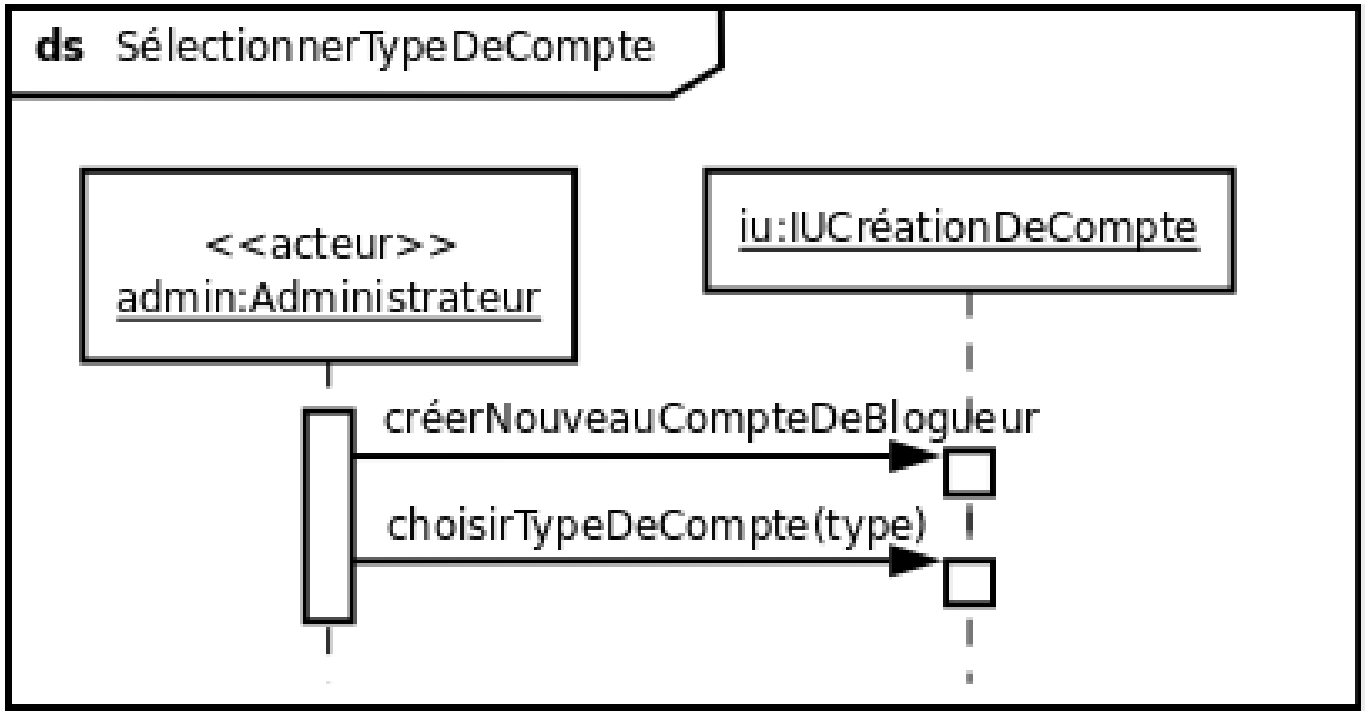
UML2 : Diagrammes de séquences

Gérer les interactions complexes avec des fragments



UML2 : Diagrammes de séquences

Gérer les interactions complexes avec des fragments



UML2 : Diagrammes de séquences

Types de fragments

Type	Paramètres	Utilité
ref	aucun	Référence une interaction définie ailleurs. Permet de gérer un grand diagramme en le découpant.
loop	itérations min, itération max, [condition_garde]	Boucle sur les interactions contenues dans le fragment, jusqu'à ce que la condition de garde ne soit plus vérifiée.
break	aucun	si les interactions de ce fragments se produisent, alors toute interaction englobante (en général loop) doit se terminer.l
alt	[cond_garde1] ... [cond_garde2] ... [sinon]	Selon la condition vérifiée, le sous-ensemble d'interactions correspondant est exécuté. if...else
opt	[cond_garde]	Les interactions de ce fragment ne sont exécutées que si la condition de garde est vérifiée. If ...