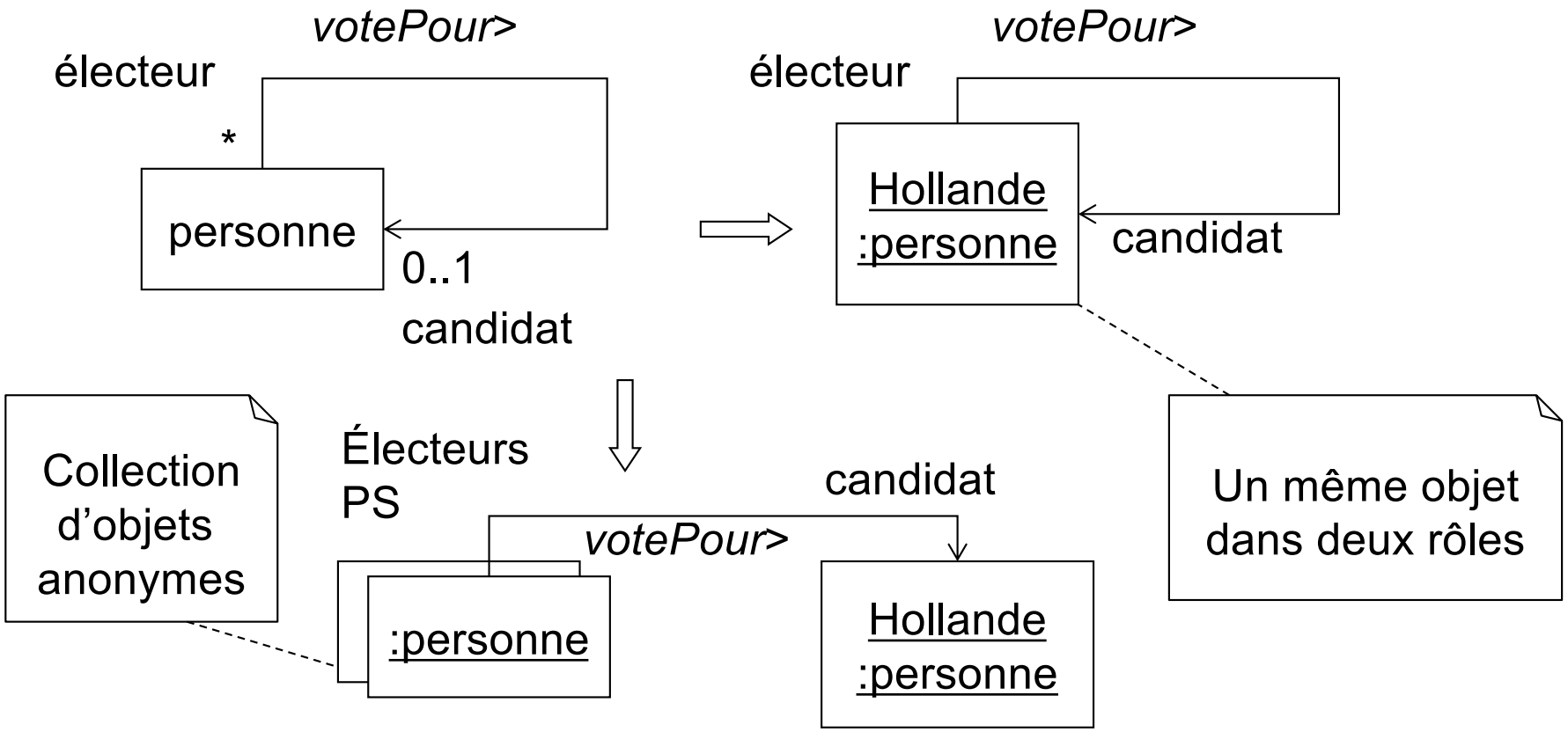
SysML/UML
Diagrammes de communication (collaboration)
Diagrammes de séquence

Diagrammes d'objets

- Instances de diagrammes de classes (donc vues statiques) destinées à éclairer certains points particuliers :

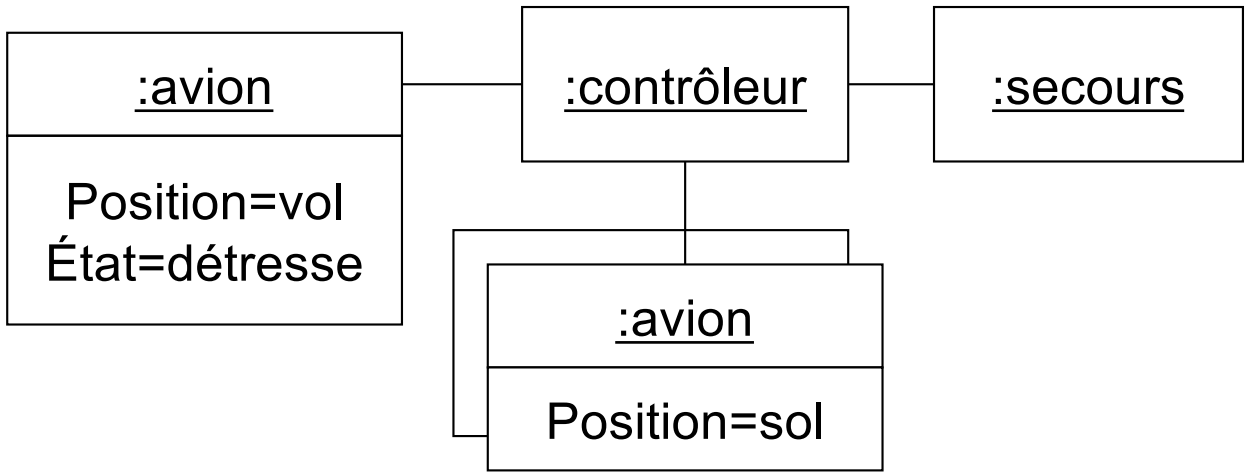


Diagrammes d'objets

- Servent essentiellement à illustrer un contexte objets liés dans un état (point de départ d'un scénario par exemple)

Diagramme de classes

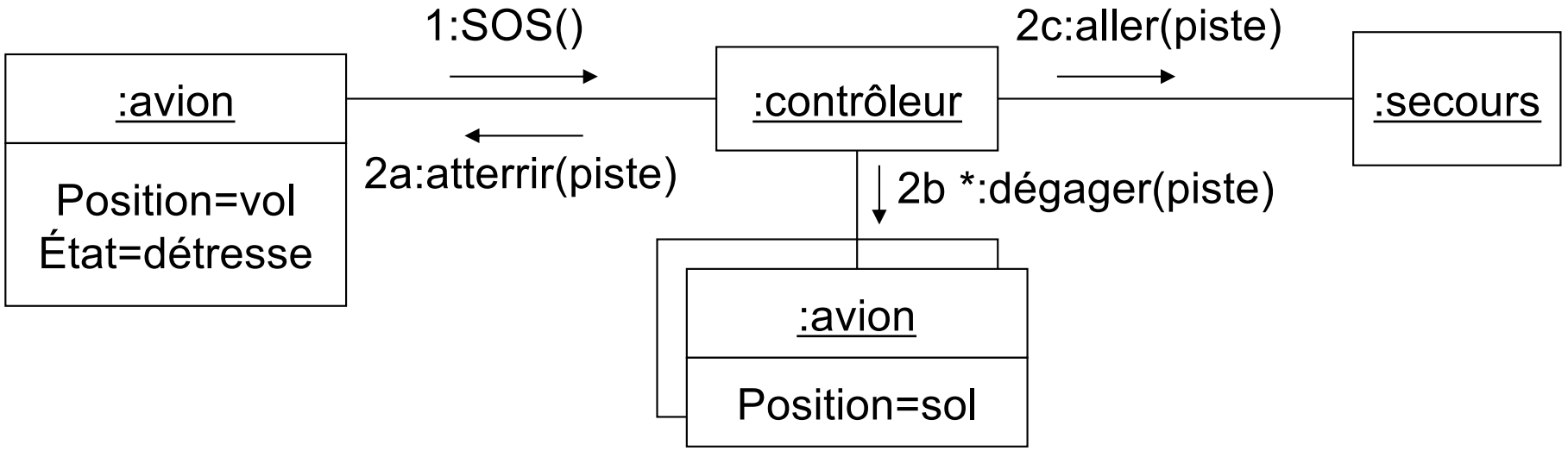
Diagramme d'objets



- SysML n'a pas retenu le diagramme d'objets qui n'y existe donc pas.

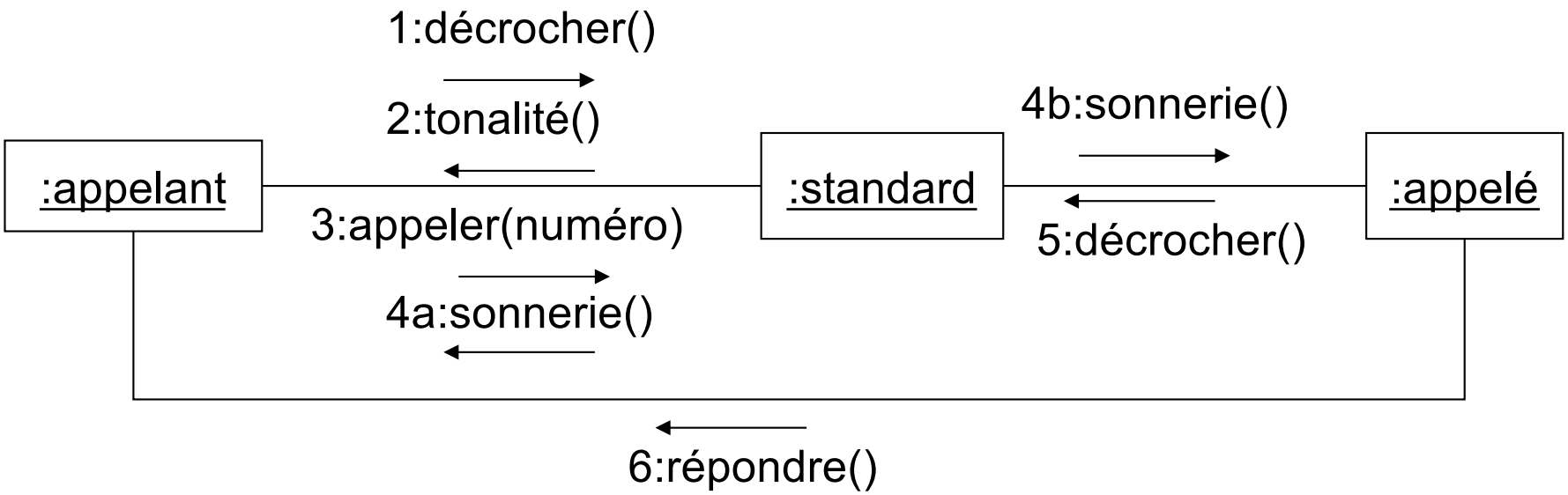
Diagrammes de collaboration (communication en UML 2.0)

- Vues dynamiques correspondant à un scénario de fonctionnement de l'application.
- Peuvent être vus comme des diagrammes d'objets enrichis des échanges de messages correspondant au scénario.
- L'aspect temporel peut être précisé par une numérotation optionnelle de l'ordre d'envoi des messages.



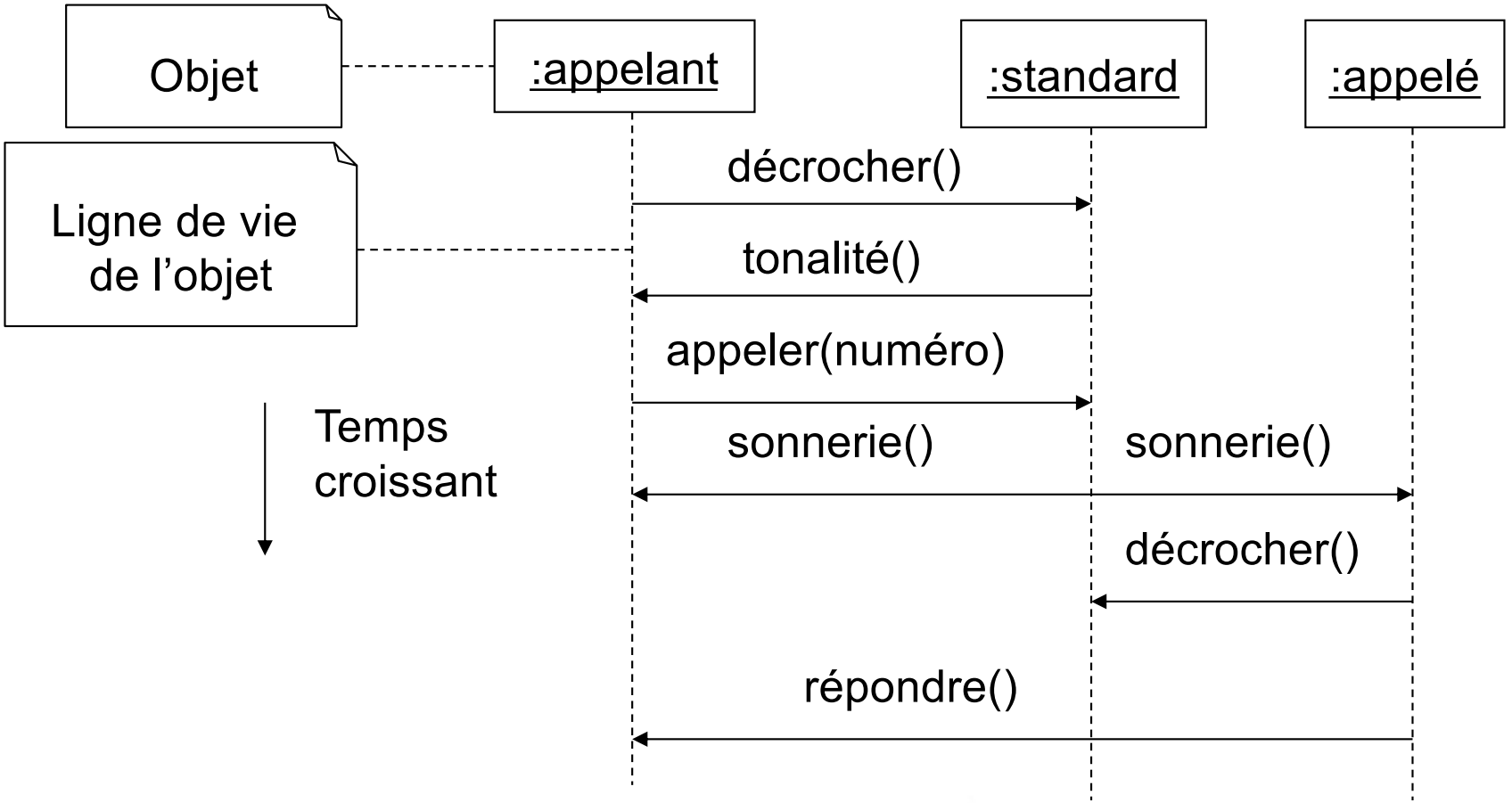
Diagrammes de collaboration (communication en UML 2.0)

- Les diagrammes de collaboration ont l'avantage de montrer simultanément la structure statique (liens) et l'aspect dynamique (échanges de messages)
- Ils peuvent par contre devenir difficilement lisibles en cas d'échanges complexes :



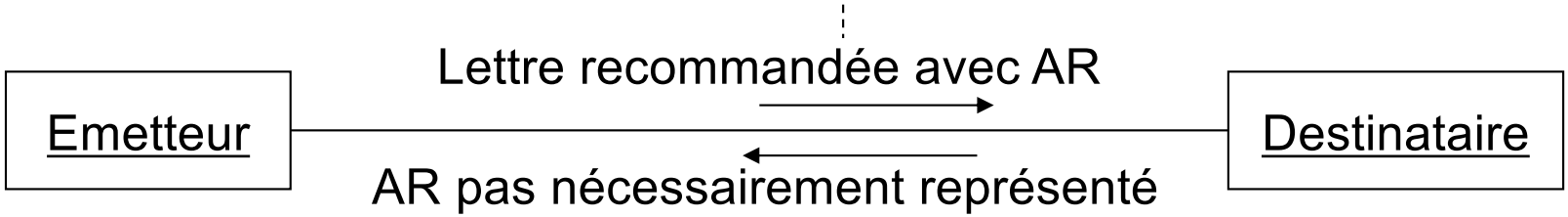
Diagrammes de séquence

- Equivalents aux diagrammes de collaboration mais privilégient l'aspect temporel :

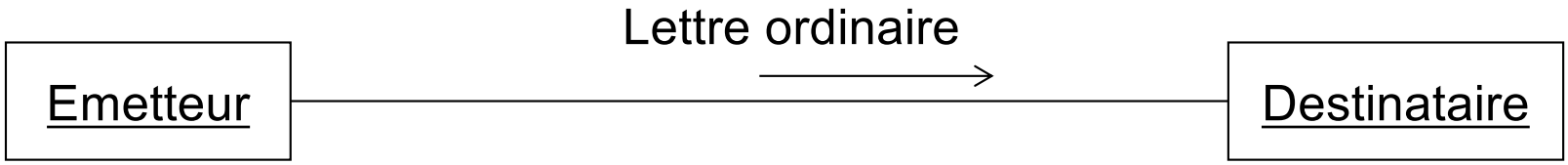


Types de messages

Message synchrone : l'émetteur est bloqué jusqu'à ce que le récepteur accepte le message (exemple : appel de procédure).
Le retour peut ne pas être explicitement représenté.

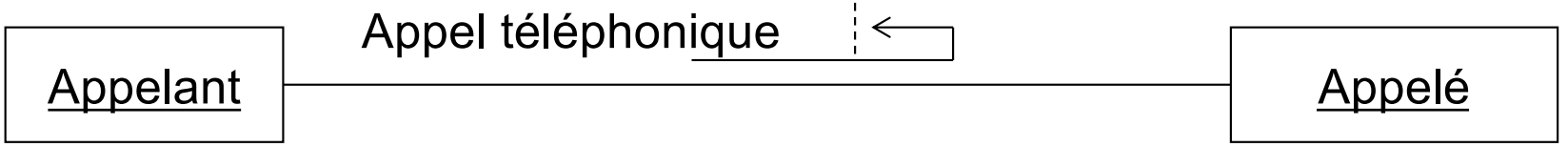


Message asynchrone : l'émetteur ne se préoccupe pas de savoir quand ni même si le message a été accepté

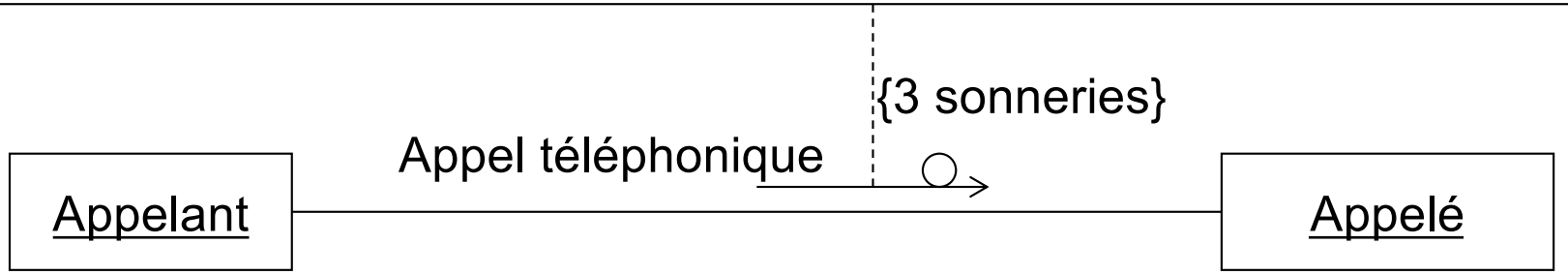


Types de messages

Message dérobant : le message n'est pris en compte que si le récepteur est disponible (l'émetteur n'attend pas)



Message minuté : l'émetteur accepte d'attendre un temps spécifié (dérobant = minuté à temps d'attente nul)



Syntaxe des messages

Liste de prédécesseurs / [condition] Numéro Itération
ValeurRetour := NomMessage(ListeParamètres)

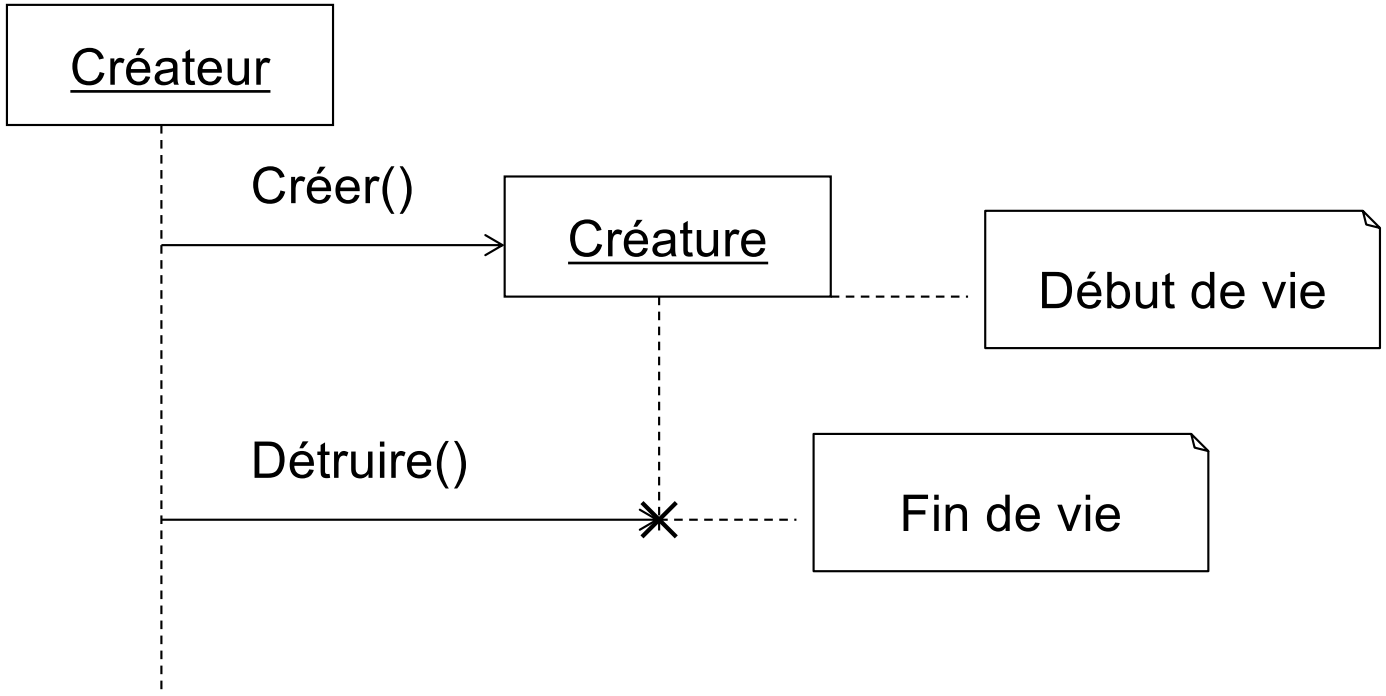
- NomMessage : seul obligatoire doit correspondre à une opération de l'objet destinataire qui peut avoir des paramètres et retourner une valeur
- Numéro : séquentialité d'envoi hiérarchisée comme les paragraphes d'un document (ex : 3.1.2), en réservant les lettres à l'expression des messages simultanés.
- Prédécesseurs : liste de numéros séparés par des virgules. L'envoi ne peut avoir lieu qu'après les prédécesseurs
- Condition : booléen conditionnant l'envoi

Syntaxe des messages

- Liste de prédécesseurs / [condition] Numéro Itération : ValeurRetour := NomMessage(ListeParamètres)
- Itération : précise l'envoi en série (*) ou en parallèle (*||) et clause d'itération optionnelle (ex: [i:=1..5])
- 1 : mouvoir()
- 1.1,1.2 / 2 : mouvoir()
- [v=0] : mouvoir()
- Vitesse:=demanderVitesse(Nomvéhicule)
- *||[i:=1..5] : mouvoir()
- 1.1 / [Heure=H] 1.2 *||[i:=1..5] : v[i]:=calcv(x,y,i)

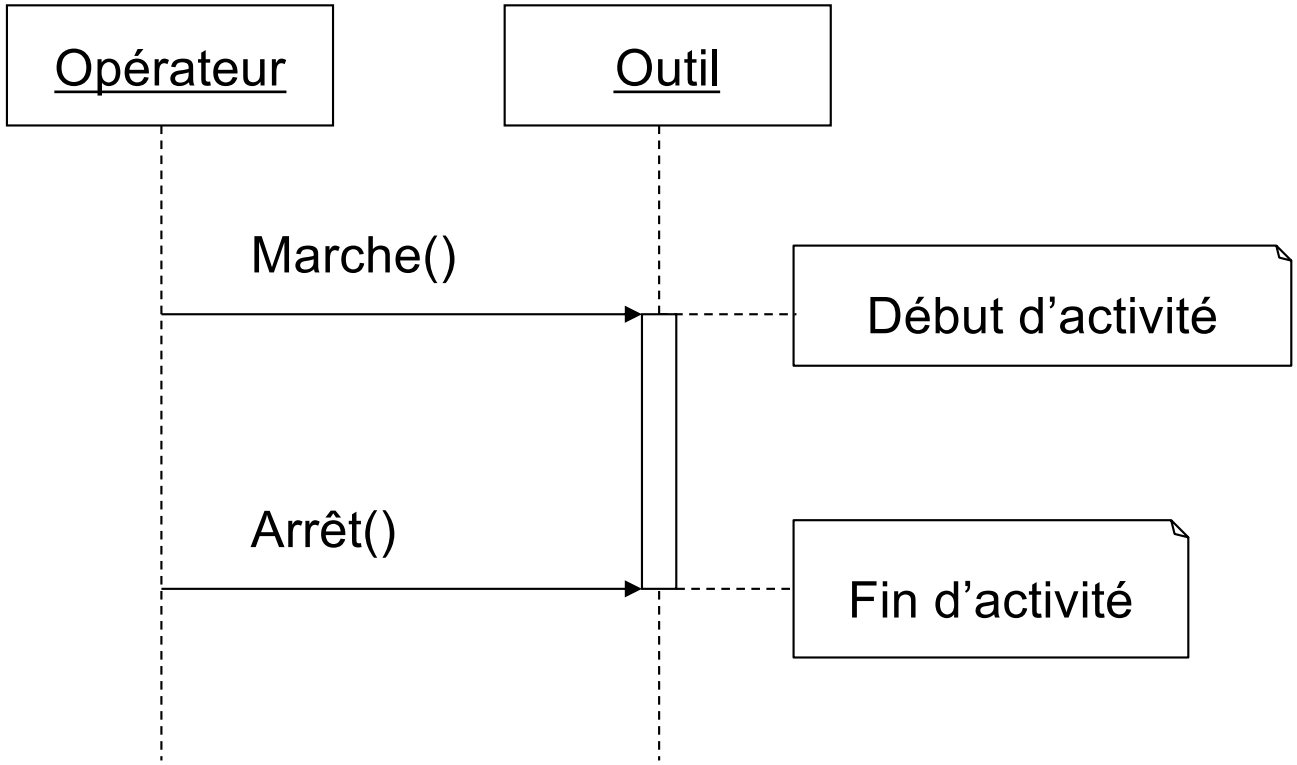
Ligne de vie des objets

- Ligne en pointillés sous les objets dans les diagrammes de séquence. Correspond à la période d'existence des objets qui peut commencer et se terminer durant la séquence :



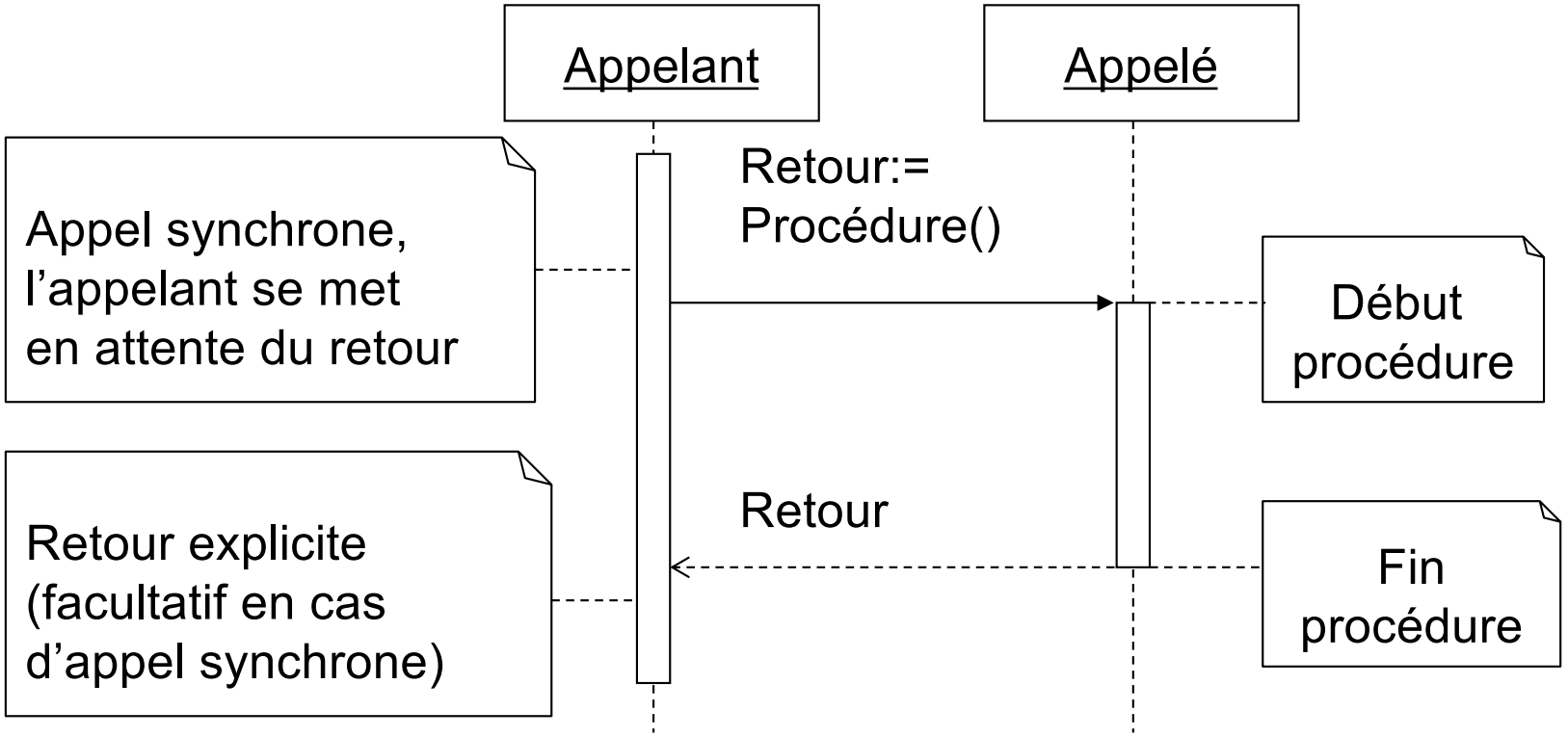
Bande d'activité des objets

- Bande rectangulaire sur la ligne de vie, permet de représenter la ou les périodes d'activité d'un objet



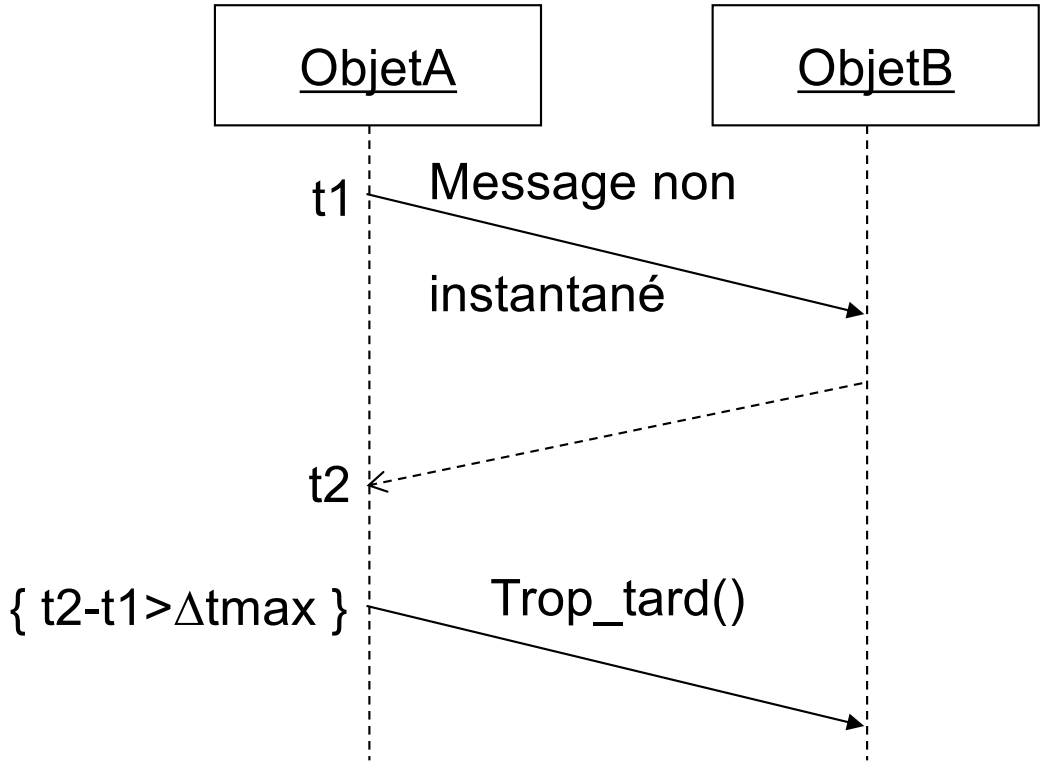
Bande d'activité : appels de procédure

- Un exemple typique est l'appel de procédure d'un objet appelant vers un objet appelé. Exemple :



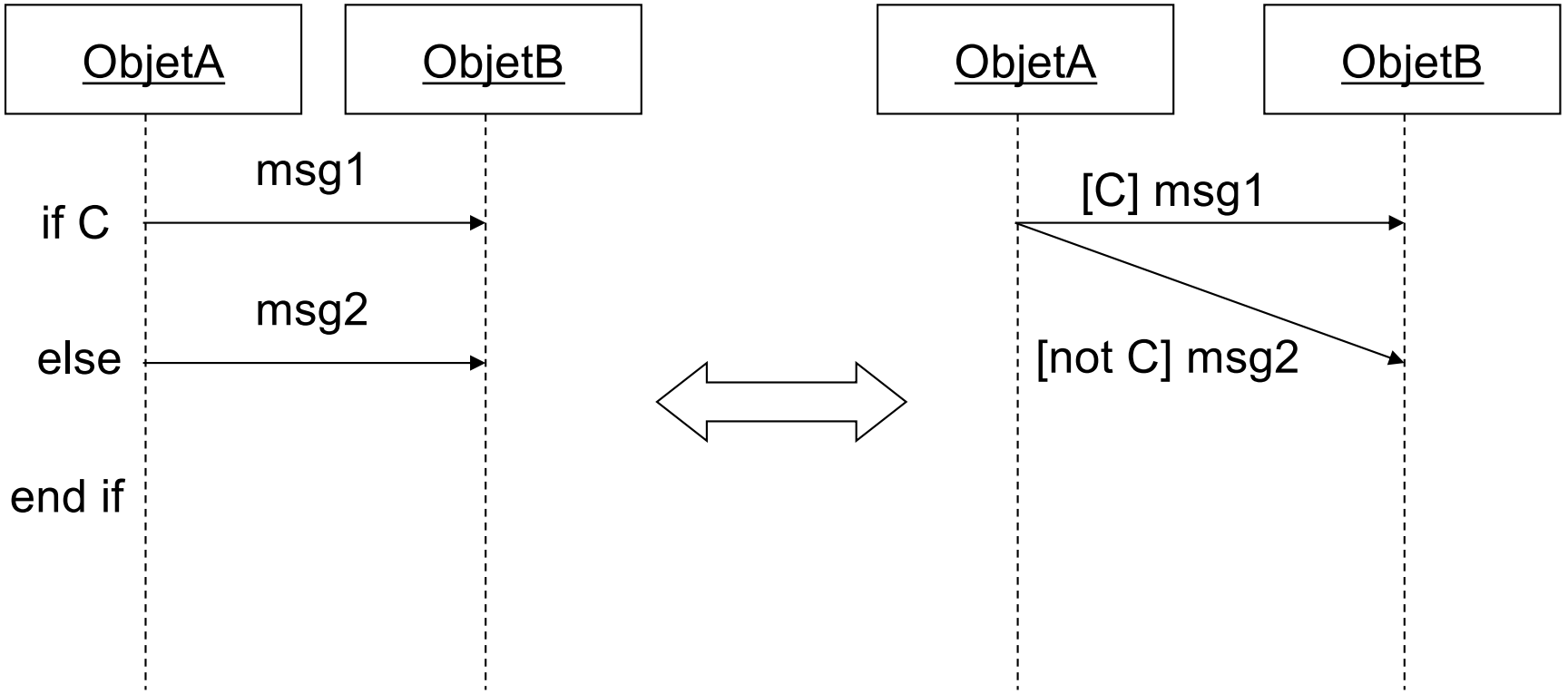
Diagrammes de séquence : contraintes temporelles

- Délai de transmissions : représenté par «message oblique»
- {Contraintes temporelles exprimées en format libre}



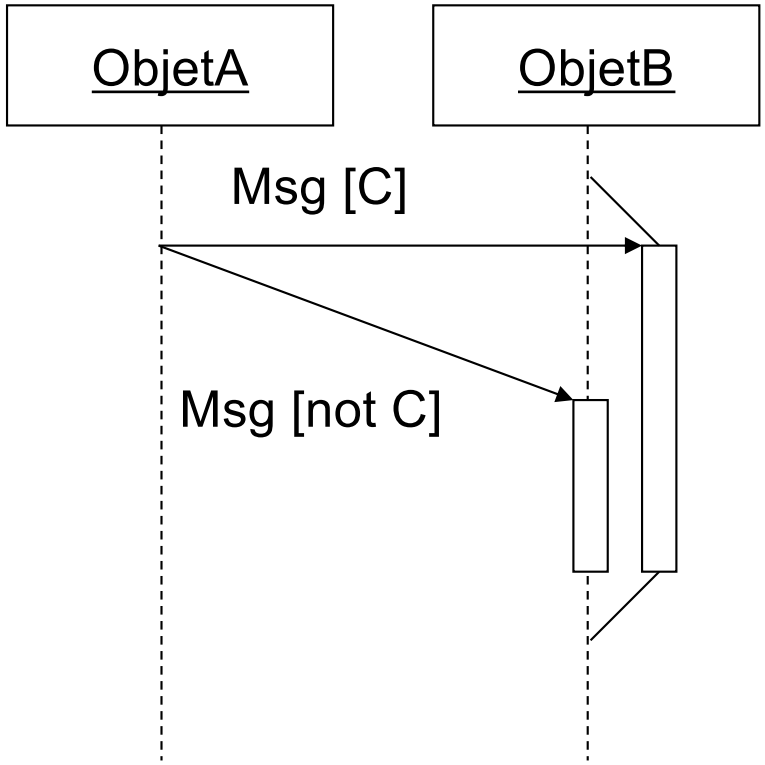
Diagrammes de séquence : Structures de contrôle

- Boucles et branchements : exprimés par du pseudo-code en marge du diagramme :



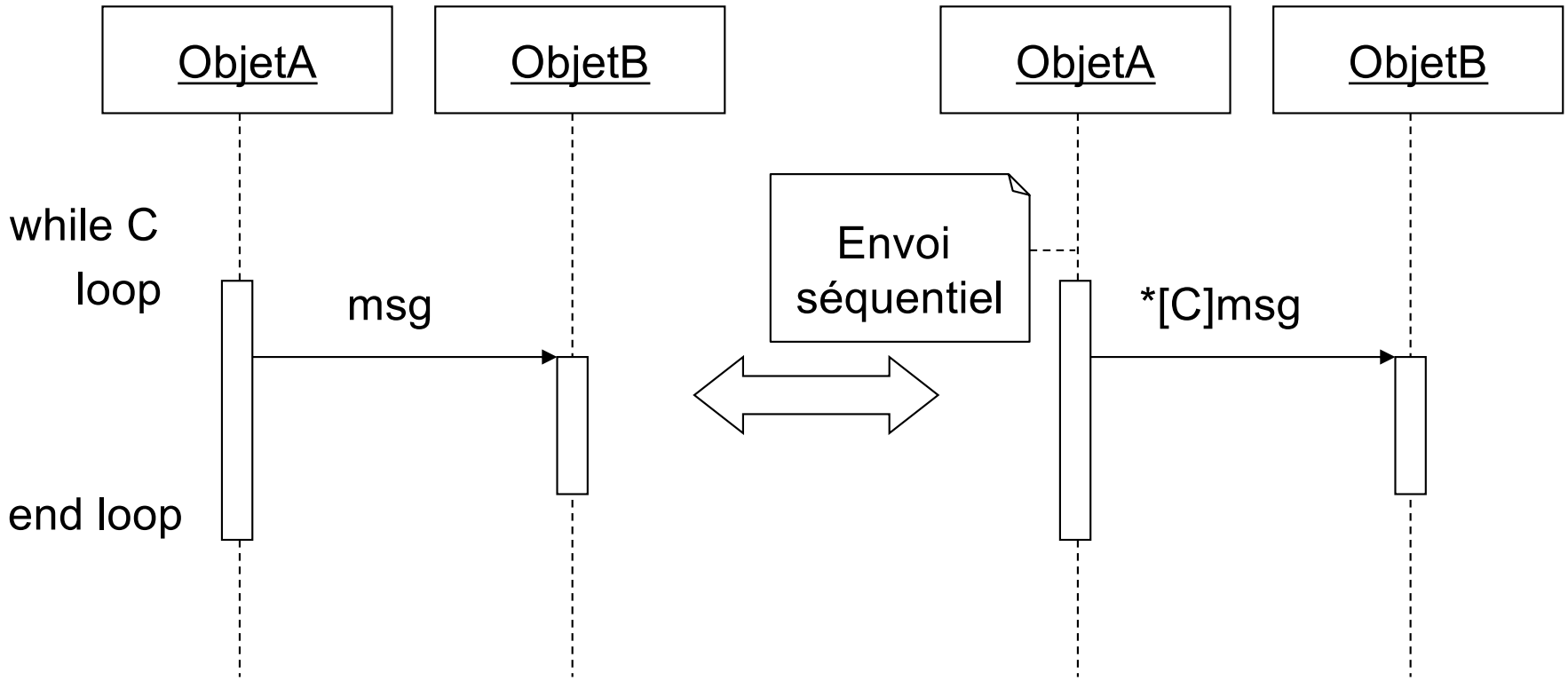
Diagrammes de séquence : Structures de contrôle

- Il est également possible de traduire un branchement conditionnel côté objet destinataire par dédoublement de sa bande d'activité :



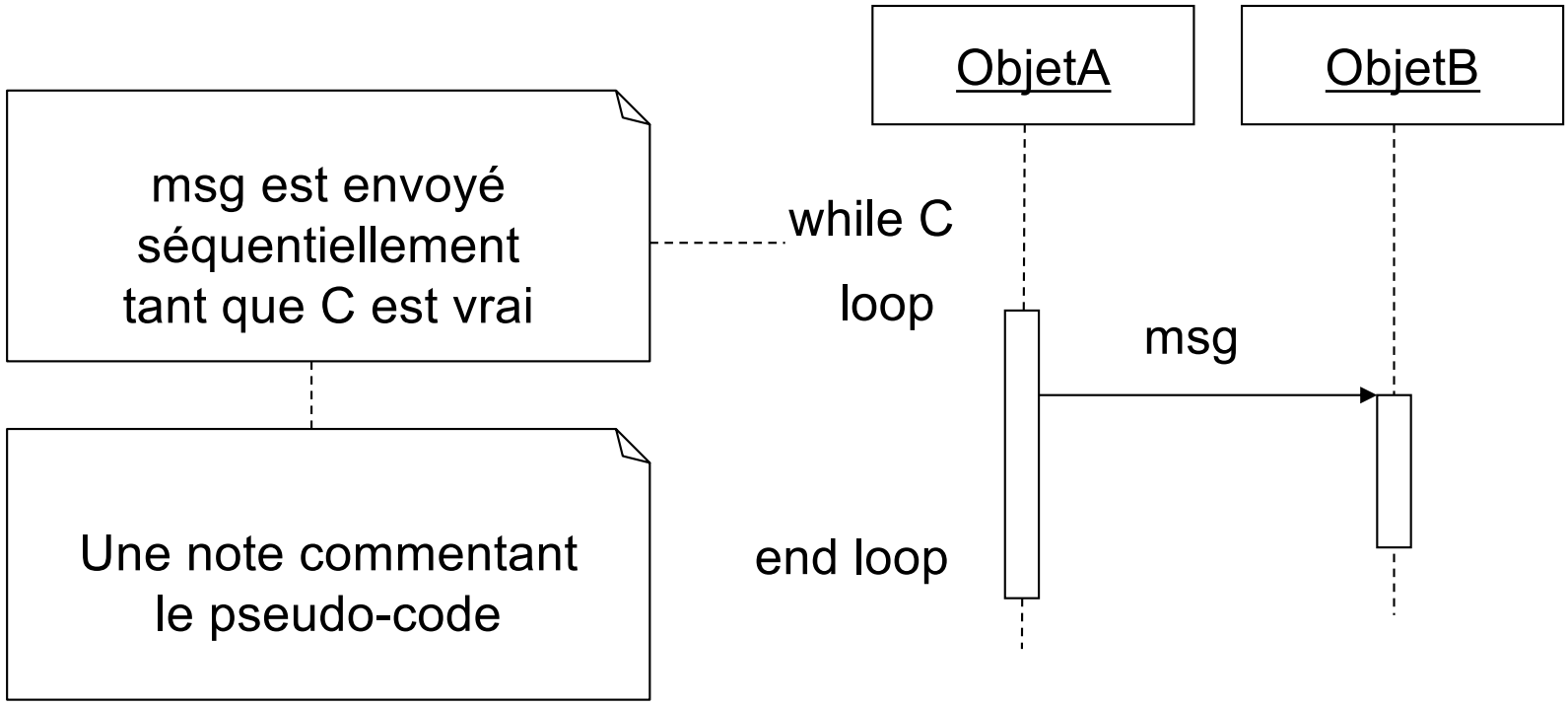
Diagrammes de séquence : Structures de contrôle

- Boucles et branchements : exprimés par du pseudo-code en marge du diagramme :



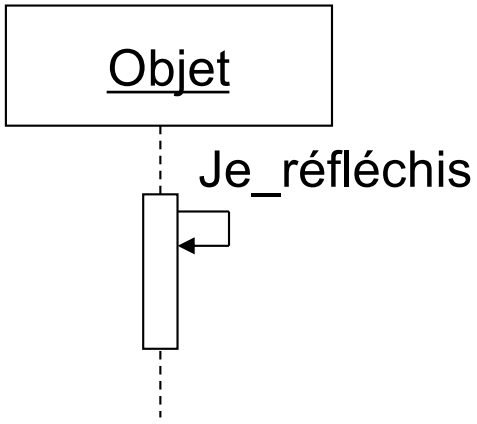
Les notes

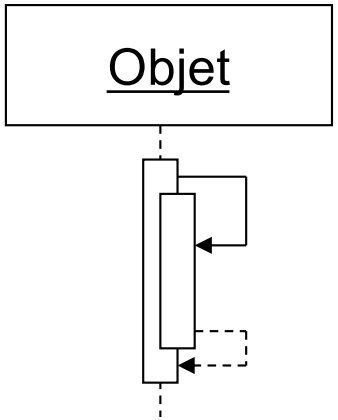
- Utilisables dans tous les diagrammes UML, rectangles à coin replié contenant un commentaire en texte libre relié par une ligne pointillée à l'élément à commenter :



Diagrammes de séquence : Messages réflexifs, récursion

- L'envoi de messages réflexifs permet d'exprimer une activité interne à l'objet (qui peut être détaillée dans un autre diagramme de séquence si l'objet est composite).
- L'envoi de messages récursifs est représenté par un dédoublement de la bande.





Adaptations SysML

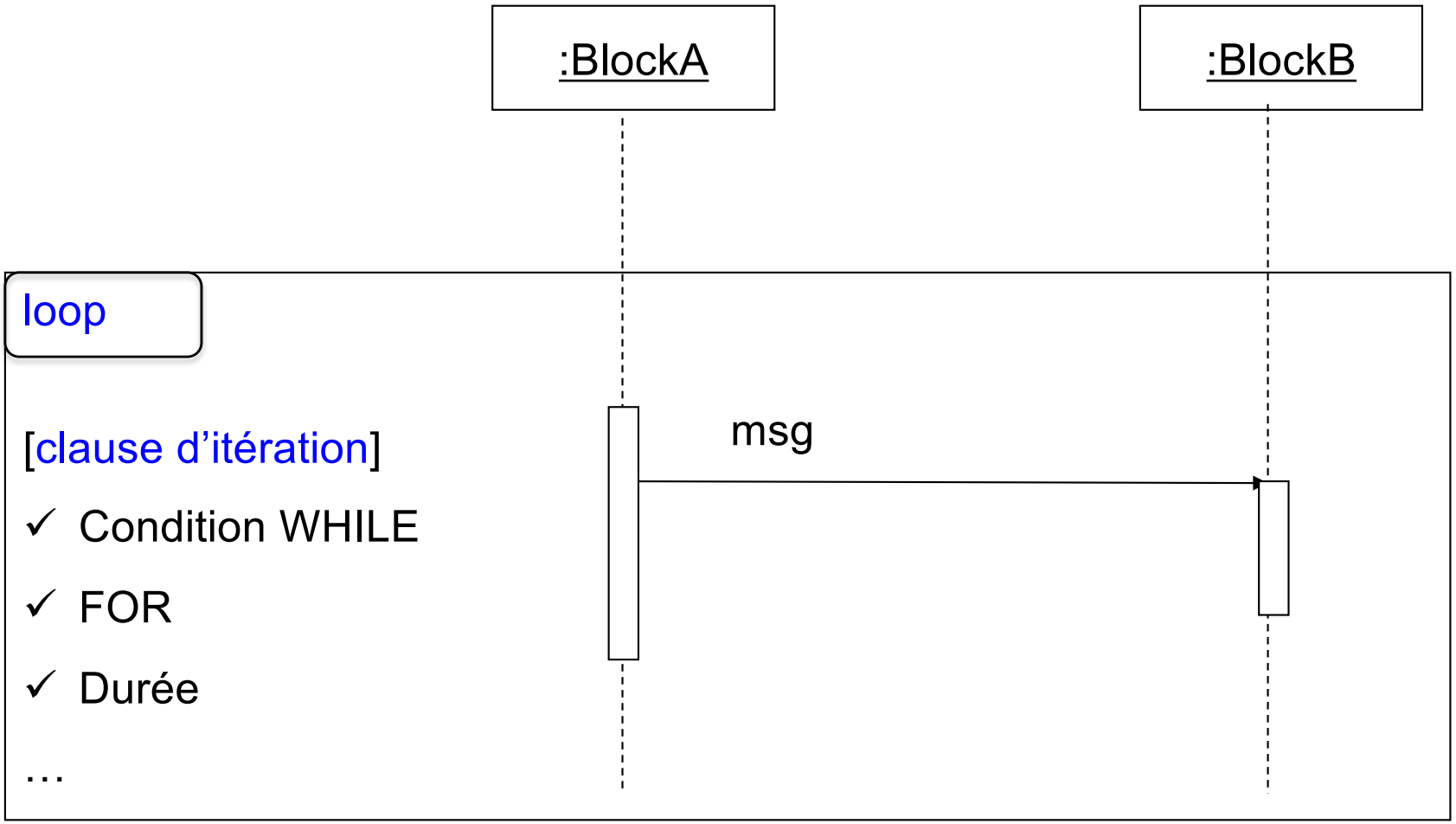
- SysML n'utilise pas le diagramme de communication
- SysML utilise en revanche le diagramme de séquence sous le nom de Diagramme de Séquence Système (DSS) (interaction acteurs / système boîte noire),
- Le DSS est exactement comme ce qui est nommé plus loin dans ce cours « scenarios use cases » (aspects dynamiques des uses cases)

Adaptations SysML

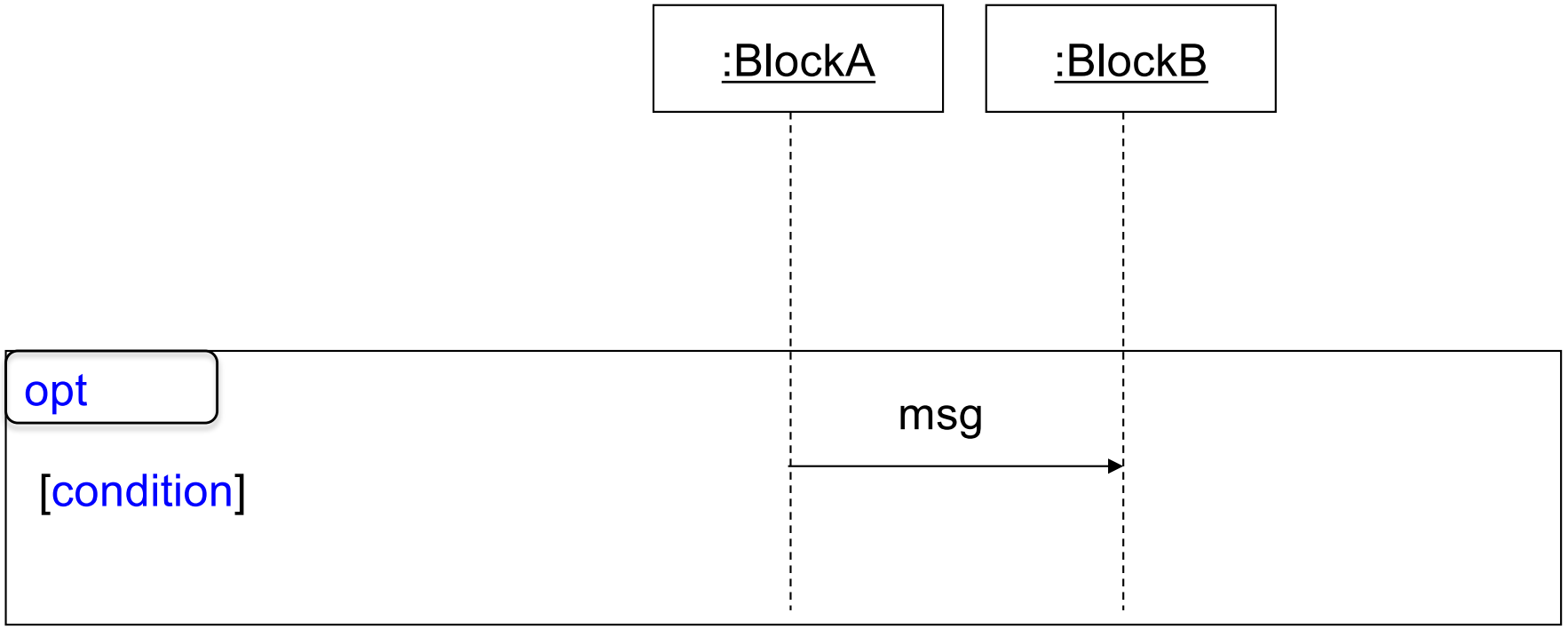
SysML permet de décomposer le diagramme en encadrant certaines parties du diagramme de séquence (parties des échanges de messages acteurs / système) par un rectangle.

- Soit pour spécifier que la partie en question est en boucle (mot clé loop), optionnelle (mot clé opt, le IF du DSS), alternatif (mot clé alt, le IF THEN ELSE du DSS), en parallèle (mot clé par pour traduire la simultanéité) : notion appelée « fragments combinés »
- Soit pour indiquer que la partie en question est décrite dans un autre diagramme dont est juste indiquée la référence (mot clé ref) : notion appelée « cadres de référence »

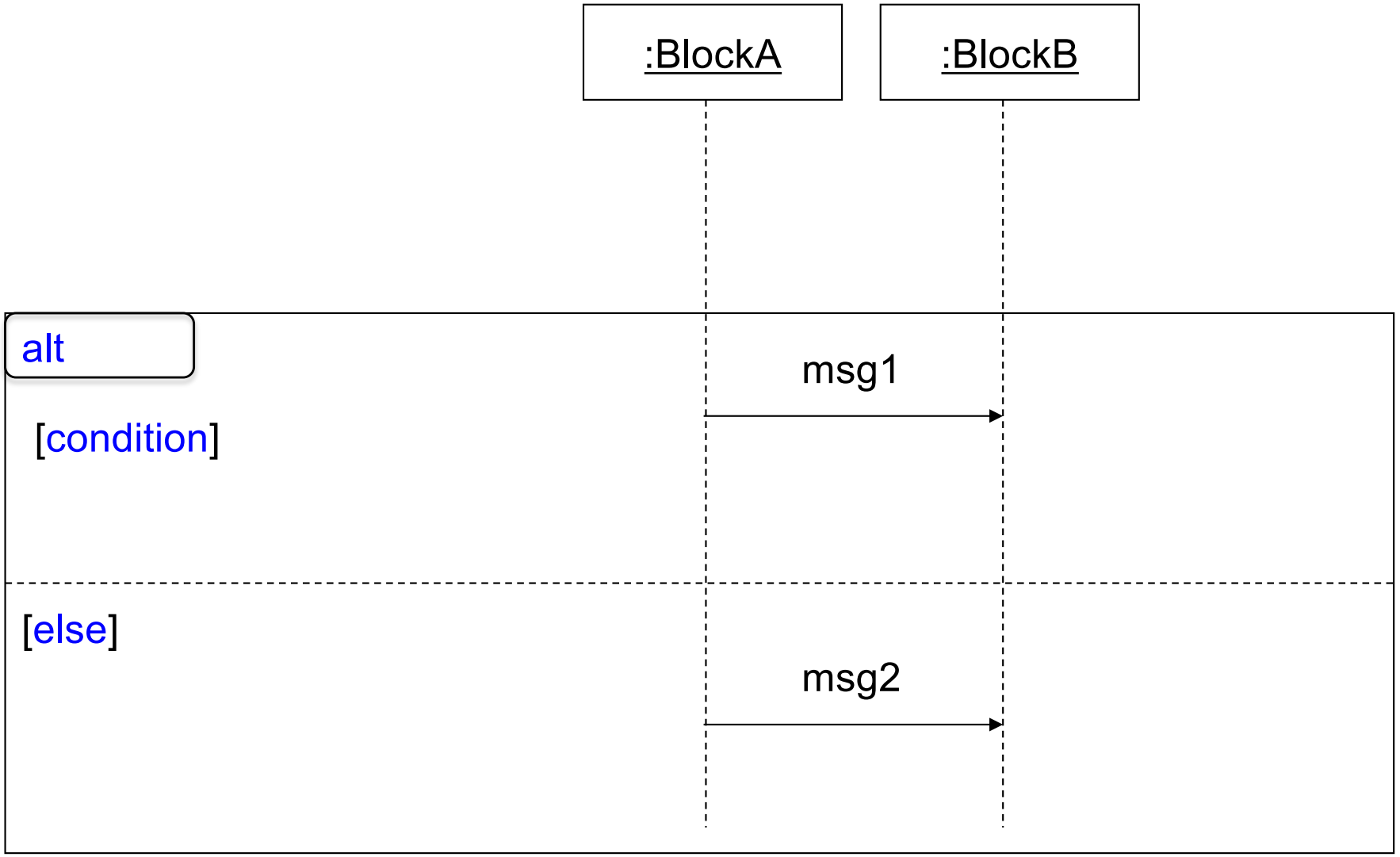
Fragment combiné loop : Boucle SysML



Fragment combiné opt : IF SysML

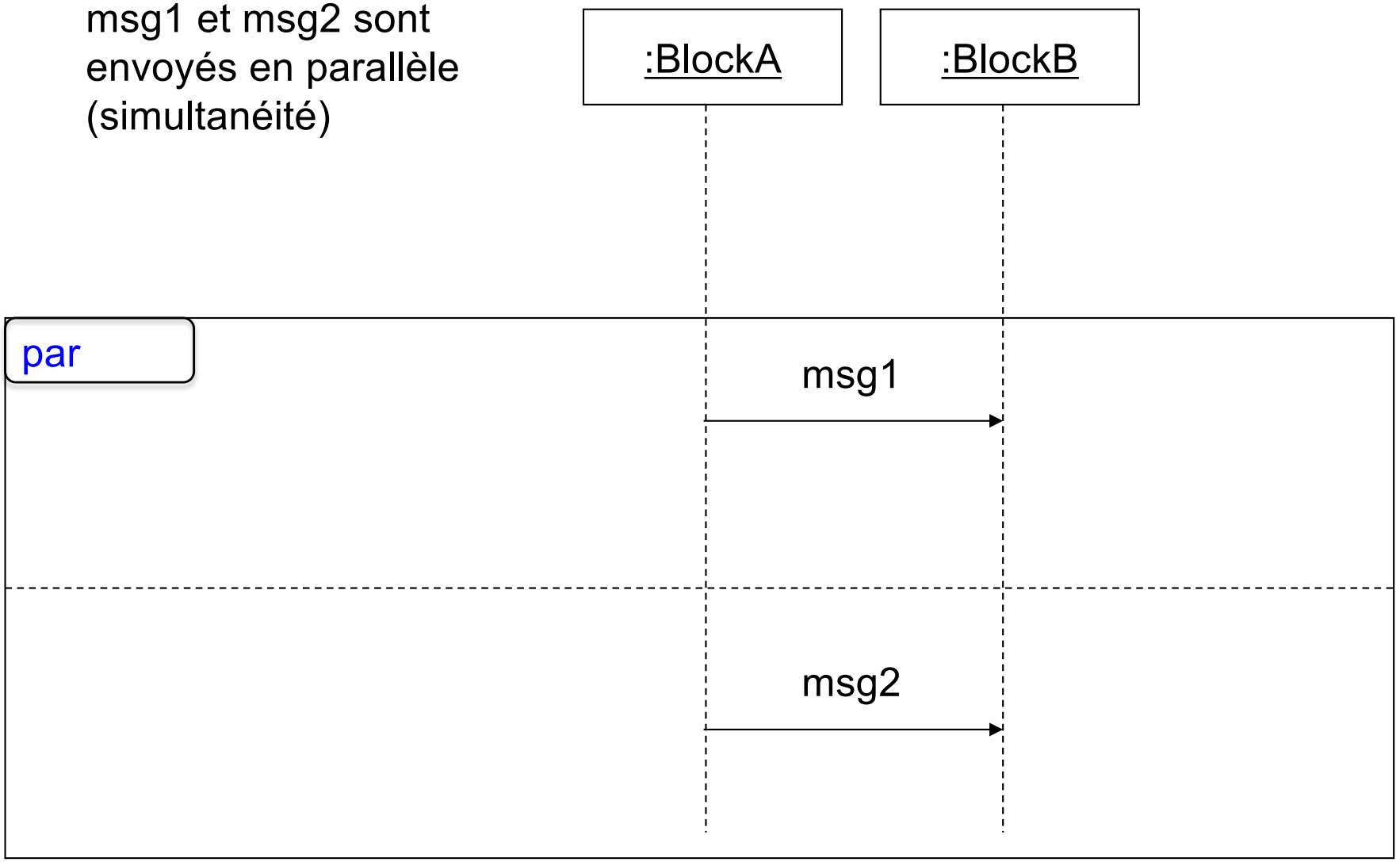


Fragment combiné alt : IF THEN ELSE SysML

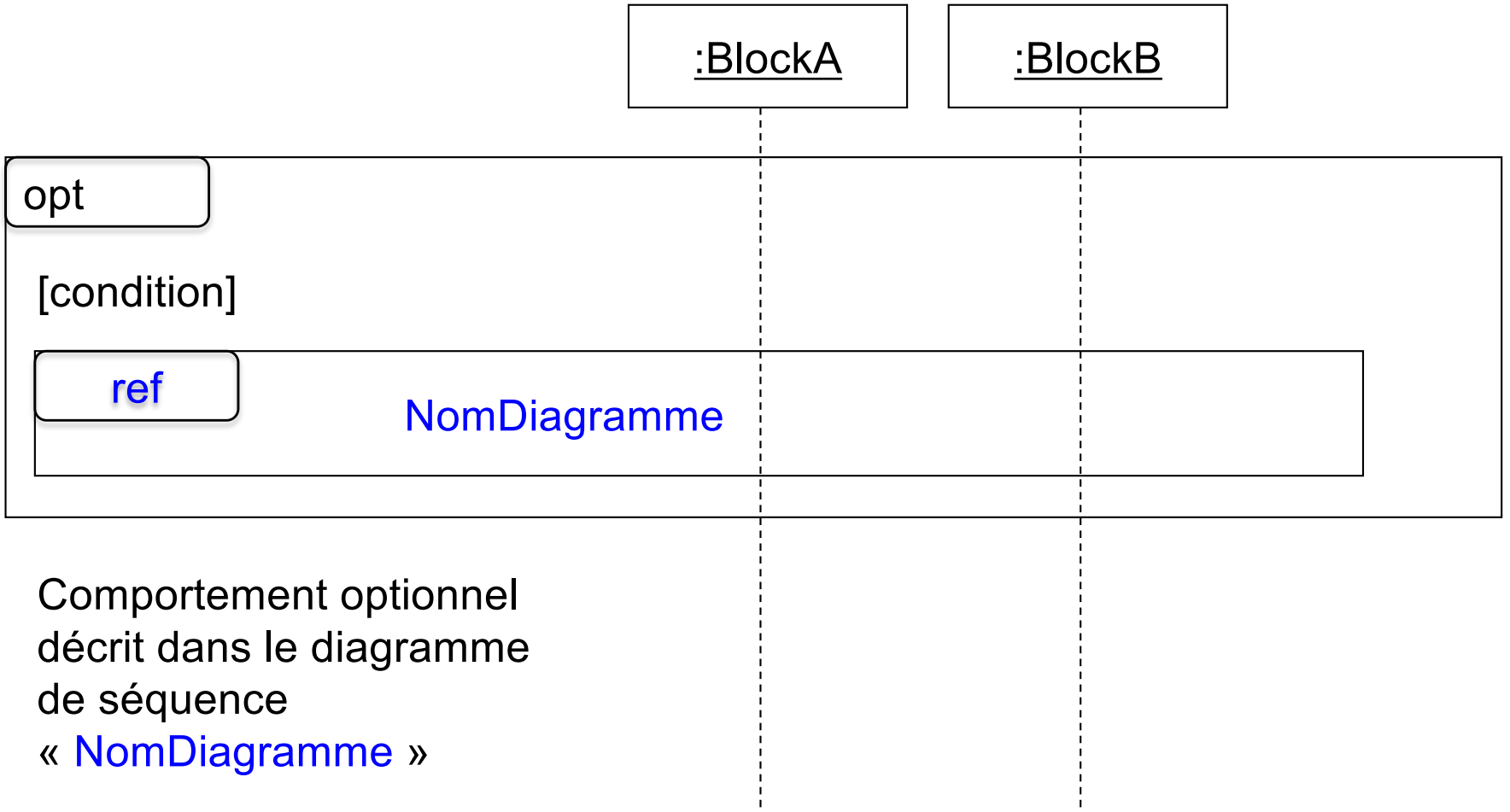


Fragment combiné par : parallélisme SysML

msg1 et msg2 sont envoyés en parallèle (simultanéité)



Cadre de référence : modularité des diagrammes de séquence SysML



Comportement optionnel décrit dans le diagramme de séquence « NomDiagramme »