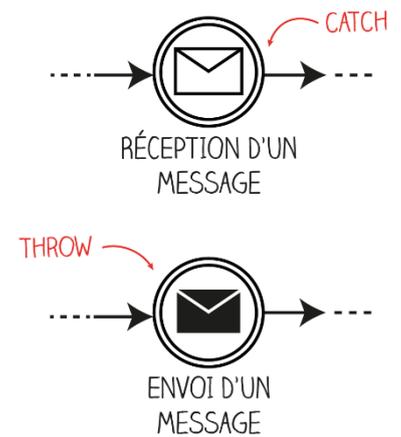
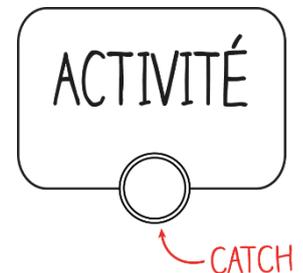


C2.5 – Les évènements (2)

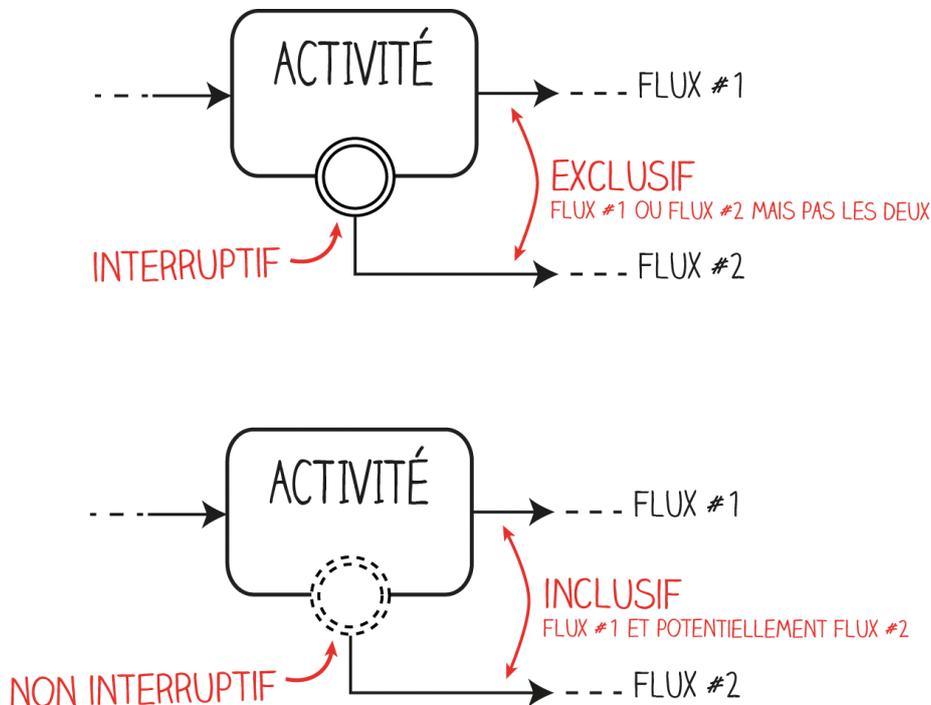
Dans notre précédent cours sur les évènements, nous avons présenté les différents types d'évènements, à savoir les évènements de départ, les évènements de fin et les évènements intermédiaires. Nous avons également vu que ces évènements pouvaient avoir différentes natures comme par exemple l'envoi ou la réception d'un message pour des messages respectivement de type « catch » (receveur) ou « throw » (lanceur). Nous avons positionné ces évènements dans le flux de séquences.



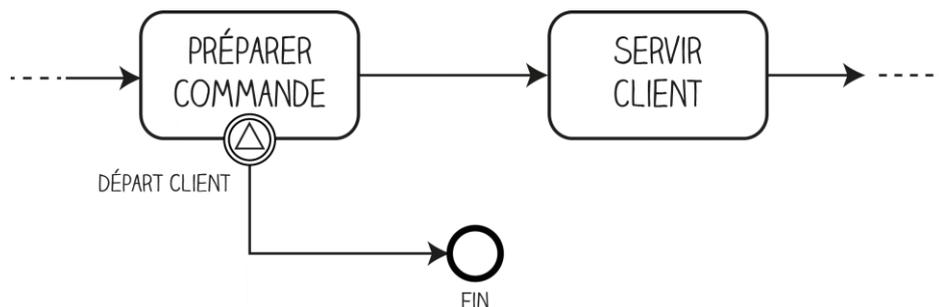
Concernant les évènements intermédiaires, il existe une autre modélisation possible. On peut en effet lier un évènement intermédiaire de type « catch » directement sur une activité. C'est ce qu'on appelle un évènement de frontière car il est positionné sur la bordure d'une activité. Cela signifie que le déclenchement de l'évènement durant l'exécution de l'activité, aura des conséquences sur le comportement du processus.



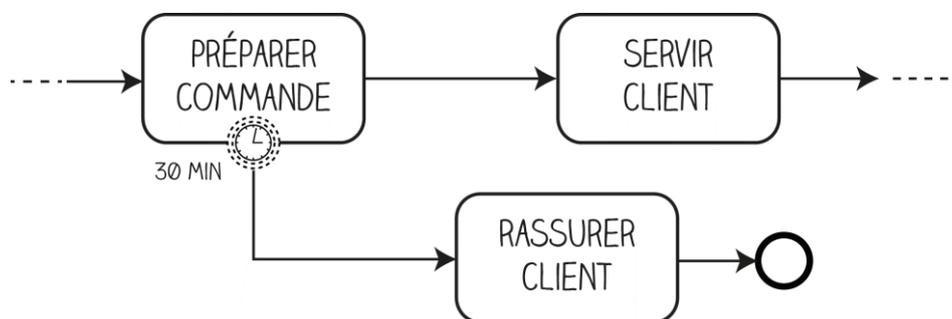
Deux possibilités : Soit le déclenchement de l'évènement interrompt l'activité ; soit le déclenchement de l'évènement n'interrompt pas l'activité. Dans ce dernier cas, les bordures de l'évènement doivent être dessinées en pointillé.



Si l'on reprend notre exemple simple de processus de service d'un client dans un restaurant, on pourrait vouloir modéliser que le départ d'un client durant la préparation de la commande interrompt le processus. On ajoute alors un évènement de frontière de type signal sur l'activité « préparer commande ».



Autre exemple, on pourrait envisager qu'au bout de 30 minutes de préparation, le serveur passe rassurer le client sur l'arrivée imminente de sa commande. Cette action n'interrompt pas l'activité de préparation de commande. On propose donc d'ajouter un évènement de type timer sur l'activité « préparer commande » en spécifiant qu'il s'agit d'un évènement qui n'interrompt pas l'activité.



Nous rappelons que seuls des évènements de type « catch », c'est-à-dire ceux reçus par le processus, peuvent être modélisés en tant qu'évènements de frontière. Les plus couramment utilisés sont les évènements de types messages, timer et règles conditionnelles.



D'autres types d'évènement sont également intéressants comme l'escalade, l'erreur, l'annulation ou encore la compensation. Mais ces derniers types sont plutôt avancés ou utilisés pour des modélisations du niveau proche de l'implémentation dans un moteur de processus, ce qui n'est pas le niveau de détail que nous avons choisi de présenter dans ce cours introductif.

