Management des Processus

BPMN - Business Process Modeling Notation

ECOLE NATIONALE POLYTECHNIQUE D'ORAN

Département Mathématiques et Informatique

Filière IMSI : 4ème année ingénieur

ram.sabri@gmail.com

M. SABRI

BPMN
Business Process Modeling Notation
Un Langage standard pour décrire les processus

- Beaucoup de sociétés font aujourd'hui du '*business process modeling*' – la représentation graphique des processus d'entreprise,
- D'autres notations avant BPMN ont été proposées pour concevoir les processus métiers d'une organisation (diagrammes de flux ou encore des diagrammes d'activités du langage UML),
- La norme en vue d'une meilleure uniformisation: la *Business Process Modeling Notation* (BPMN).

Présentation de BPMN

La première version de la norme BPMN, a été proposée par le Business Process Management Initiative (BPMI) en 2004,

Elle a ensuite été reprise par l'OMG (Object Management Group).

Objectifs

- Langage commun accessible (Expression commune des processus),
- Favoriser l'implémentation et l'exécution des processus métiers.

A qui s'adresse cette norme ?

BPMN s'adresse à tous ceux qui s'intéressent à la compréhension et à l'optimisation des processus métiers :

- Les consultants et auditeurs en stratégie d'entreprise,
- Les analystes métier ou de processus,
- Les concepteurs de processus métier,
- Les architectes ou urbanistes de systèmes d'information,
- Et également tous les managers intermédiaires et supérieurs !

Pourquoi utiliser la norme BPMN en particulier ?

Le BPMN a tendance à faire l'unanimité dans la gestion de processus et pour plusieurs raisons :

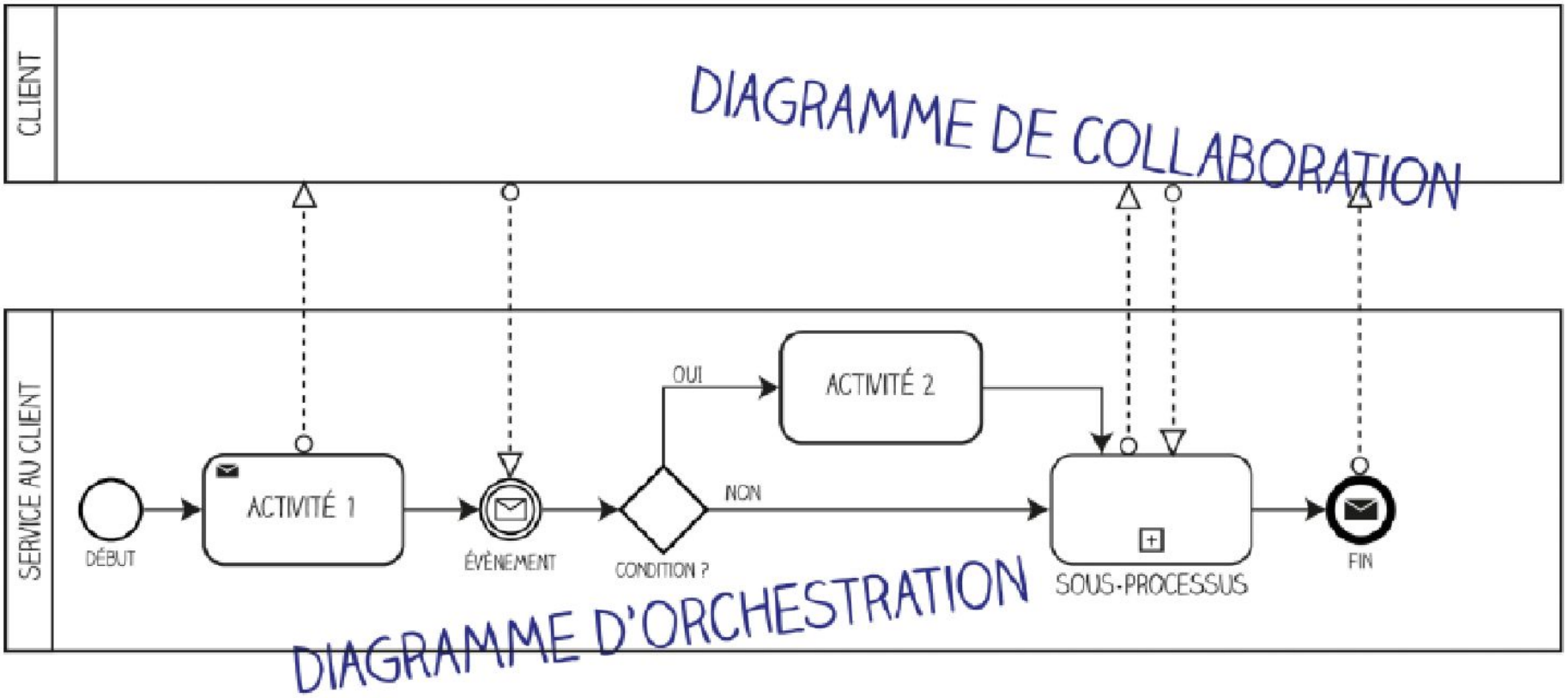
- Une notation ciblée métier : particulièrement efficace pour exprimer les métiers,
- Une norme internationale : elle est standardisée et neutre, garantie par l'OMG,
- Une norme « ouverte » ou non propriétaire,
- Une norme vivante.

Les différents modèles

Pour répondre à différents besoins de modélisation de processus, BPMN définit une suite de 4 modèles :

- **Diagramme d'orchestration** : ce diagramme correspond à la définition de la séquence d'un processus, du début à la fin, avec les différents concepts à savoir : les flux, les évènements, les activités et les passerelles.
- **Diagramme de collaboration** : il permet de représenter graphiquement les interactions entre un processus et d'autres processus, internes ou externes à l'organisation, en spécifiant les messages échangés.

Les différents modèles



Les différents modèles

- **Diagramme de conversation** : ce diagramme met l'accent sur les communications entre participants (Nœuds de conversation),
- **Diagramme de chorégraphie** : il est utilisé pour analyser la façon dont les participants échangent des informations afin de coordonner leurs interactions. Vous pouvez utiliser un diagramme de chorégraphie afin de développer et d'analyser en détails l'échange des messages associés à un nœud de conversation dans un diagramme de conversation.

Les différents modèles

Pour décrire ces différents modèles, BPMN définit des spécifications formelles avec un ensemble de ***symboles graphiques***, ***une syntaxe***, ***une sémantique*** et des ***règles d'usages***.

D'une certaine façon, on peut ***assimiler BPMN*** à un ***langage***, avec une ***syntaxe***, une ***grammaire***, et un ***vocabulaire*** qui donne un ***sens global*** en fonction de son expression.

BPMN garantit ainsi l'homogénéité dans l'expression et l'interprétation des modèles.

Concepts de base

- Ils incluent les ***activités***, ***branchements*** et ***événements***, et les ***flux séquentiels*** qui les lient.
- Chacun de ces éléments propose plusieurs types qui peuvent être connectés dans une séquence.

- **Activités**
- **Flux séquentiels**
- **Événements**
- **Branchements**

Concepts de base

Activités

Une activité est une action, une unité de travail réalisée aux cours d'un processus, avec un début et une fin bien identifiée.

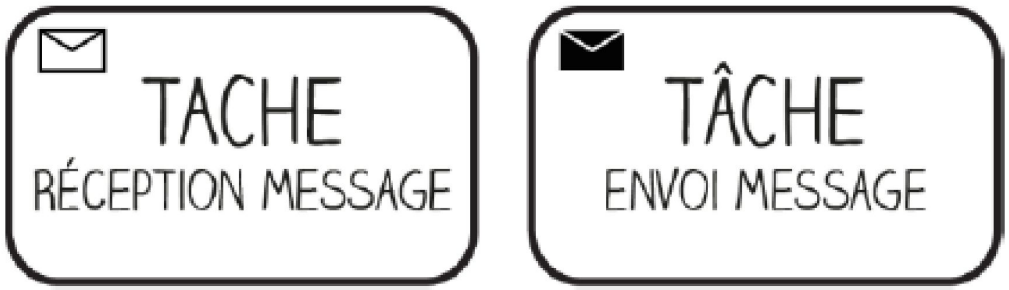
- Tâches,
- Sous processus.

Concepts de base
Activités - Tâches

- Une tâche est un type d'activité qui ne peut être décomposée, c'est-à-dire qu'il n'est pas possible de définir un niveau de détail plus fin. On parle d'unité ***élémentaire*** ou ***atomique***.
- Une tâche étant dédiée à une action bien particulière, il est possible d'en spécifier sa ***nature***.
- Nous pouvons ajouter des ***pictogrammes*** en haut à gauche de l'activité pour définir le type de la tâche.

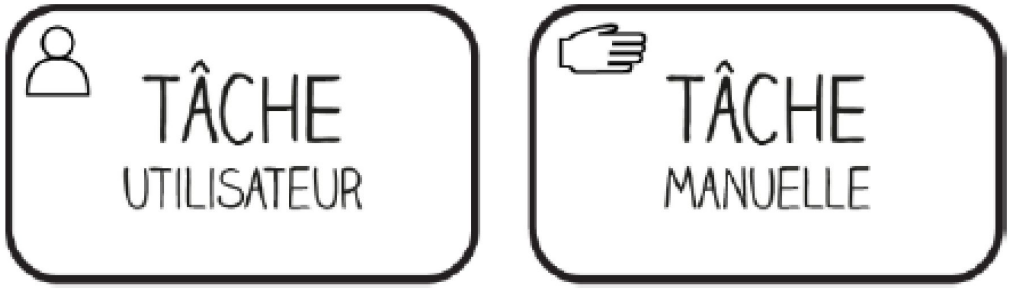
Concepts de base
Activités - Tâches

- Tâche de ***réception*** qui spécifie que l'on reçoit un message d'un utilisateur extérieur
- Tâche d'***envoi*** qui spécifie que l'on envoie un message à un utilisateur externe au processus.



Concepts de base
Activités - Tâches

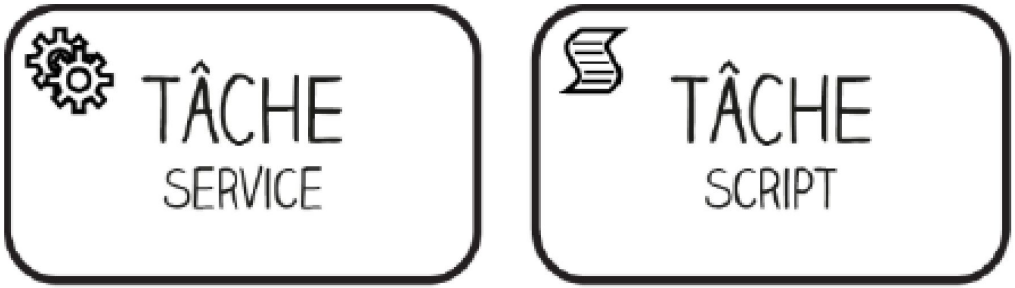
- Tâche ***utilisateur*** précise que la tâche est réalisée par un acteur humain mais interagissant avec une application informatique,
- Tâche ***manuelle*** qui est réalisée exclusivement par un acteur humain.



Concepts de base

Activités - Tâches

- Tâche de ***service*** est une tâche automatisée, c'est à dire sans intervention humaine. L'application informatique déclenchée est vue comme un service demandé. On ne connait pas le contenu de l'application, on ne connait que les résultats produits.
- Tâche de ***script*** est également une tâche automatisée mais dont le comportement a été construit spécifiquement pour la gestion du processus. C'est donc un type de tâche qui ne concerne que ceux qui souhaiteraient à terme créer un moteur pour la gestion automatique de leur processus.

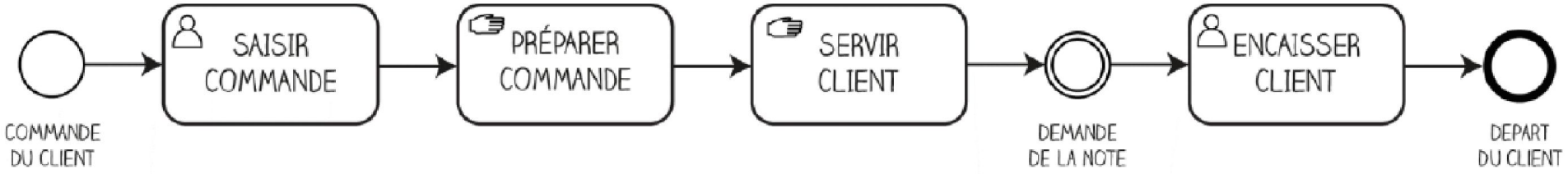


Concepts de base
Activités - Tâches

- Tâche de ***règle de gestion*** permet d'indiquer que l'action de la tâche est d'appliquer une règle métier pour prendre une décision.

Concepts de base
Activités - Tâches

- Exemple :



Dans notre processus simple de service au client dans un restaurant, la saisie de la commande serait une tâche utilisateur, tout comme la tâche de l'encaissement. Les tâches de préparation de la commande et de service du client seraient des tâches exclusivement manuelles.

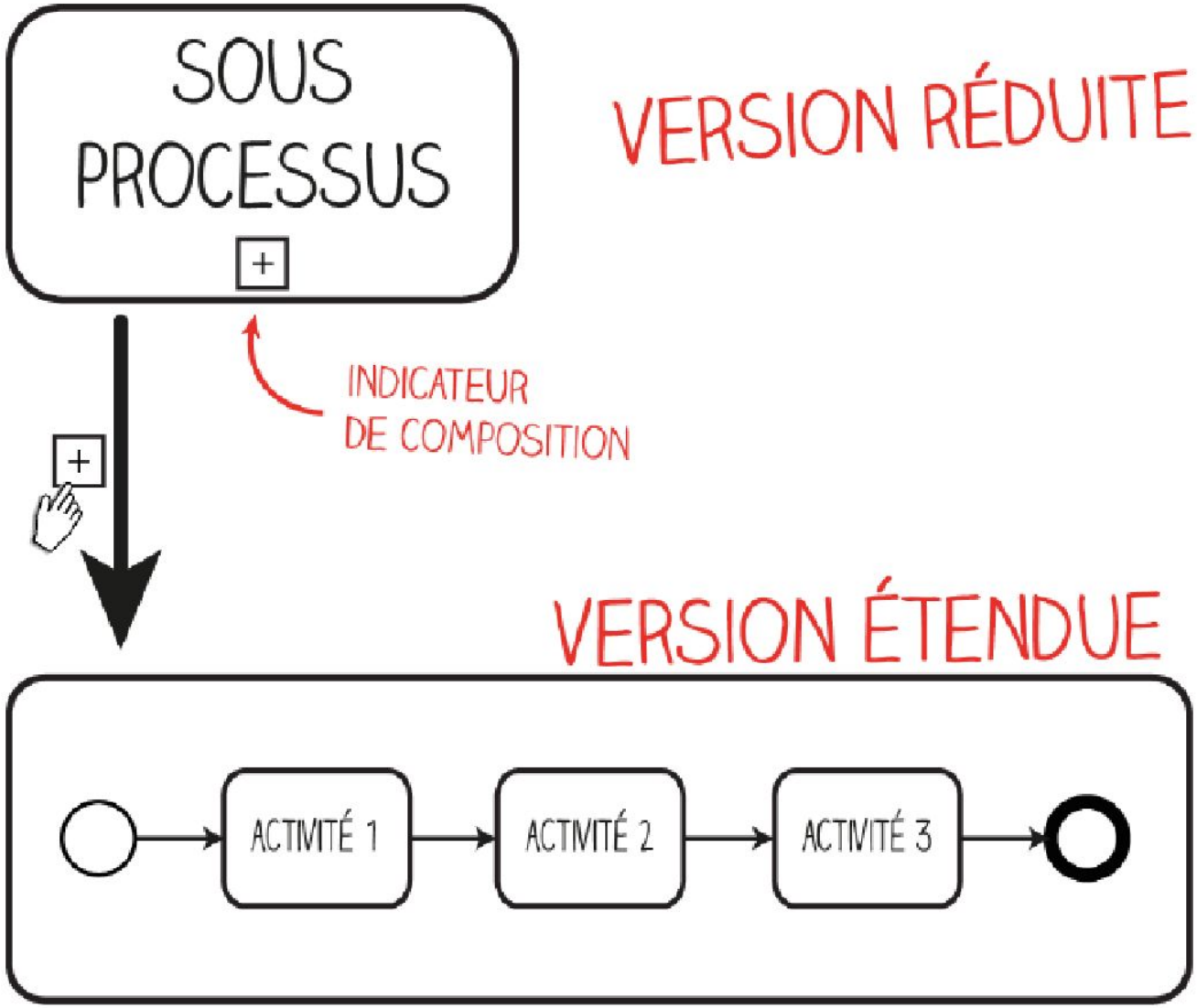
Concepts de base

Activités – Sous-processus

- À l'inverse de la tâche, un sous-processus est une activité ***composée***, c'est-à-dire une activité qui peut elle-même être décrite suivant une séquence d'activités.
- Le sous-processus peut être représenté de deux façons : soit dans une **version réduite** ; un marqueur carré avec le signe + est alors inscrit sur le sous-processus. Ce signe ne doit pas être confondu avec la notion de parallélisme que l'on a pu voir avec les évènements ou les passerelles. Il permet simplement de spécifier le fait que cette activité est composée.
- Dans le diagramme, le sous-processus peut également être affiché dans sa **version étendue**. La forme de l'activité est alors élargie afin de laisser la place d'inscrire la séquence du sous-processus à l'intérieur.

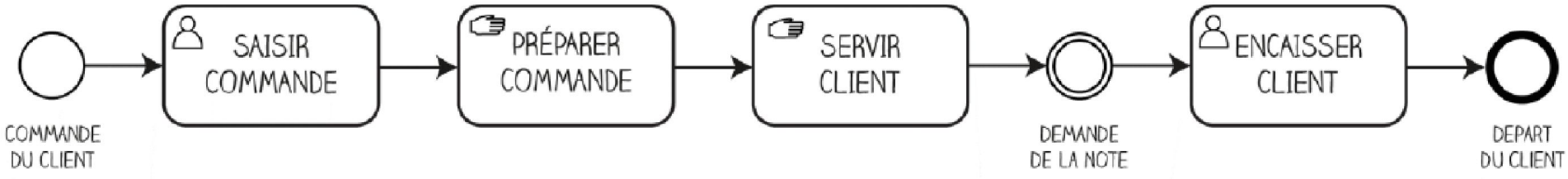
Concepts de base

Activités – Sous-processus



Concepts de base

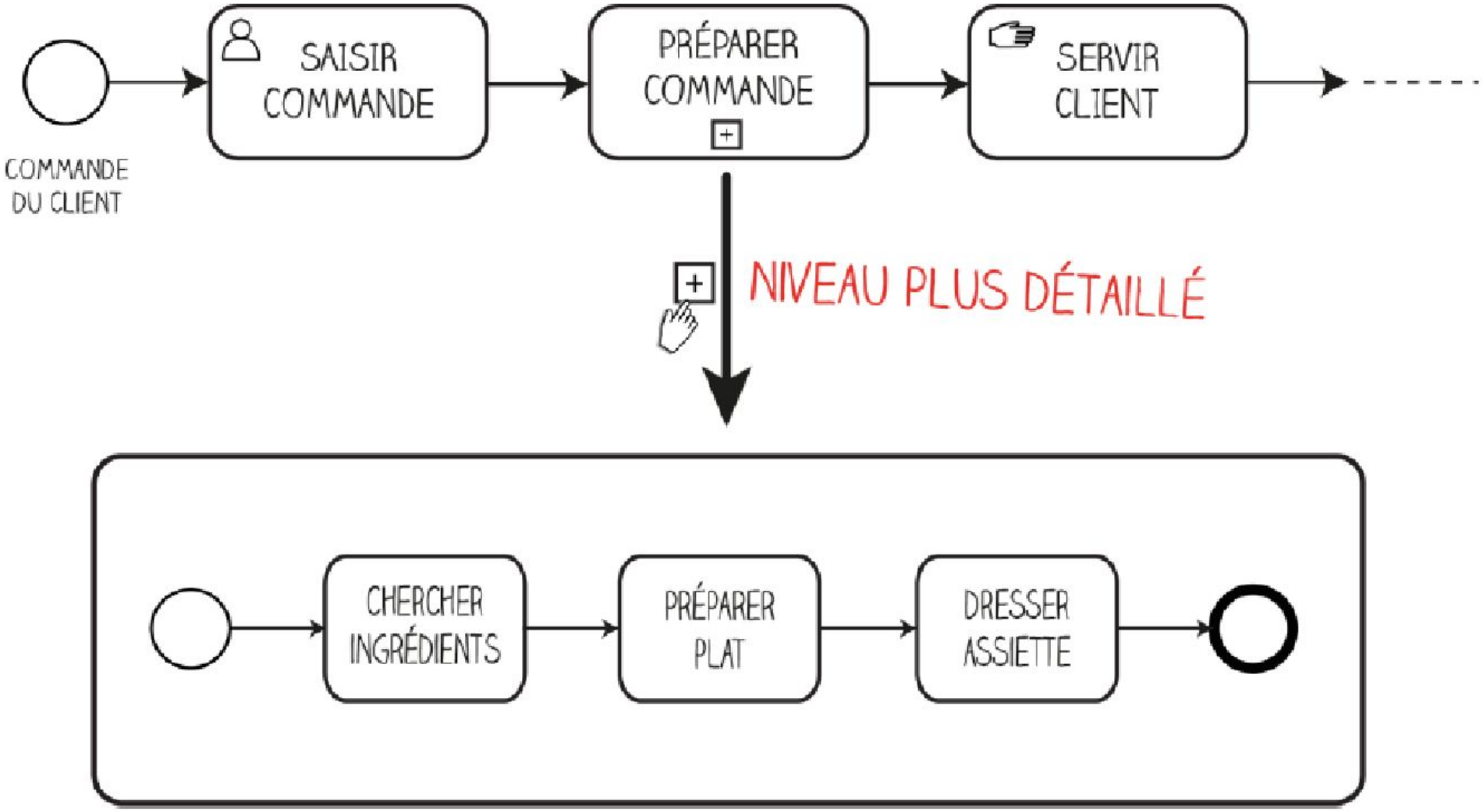
Activités – Sous-processus



- Dans notre exemple, on pourrait facilement dire que l'activité « *préparer commande* » est un sous-processus, c'est-à-dire qu'elle contient elle-même une séquence d'activités.
- Si l'on étend l'affichage du sous-processus, nous pourrions obtenir un flux d'activités composé d'un évènement de départ, de différentes tâches « *chercher ingrédients* », « *préparer plat* » et « *dresser assiette* », flux qui se termine par un évènement de fin.

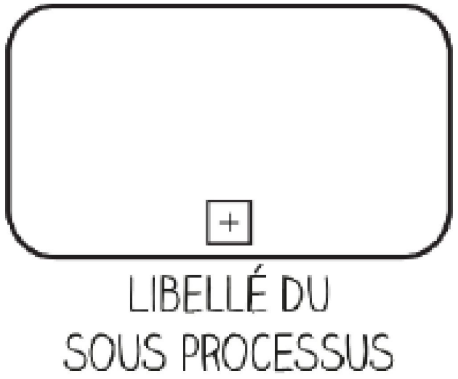
Concepts de base

Activités – Sous-processus



Concepts de base
Activités – Sous-processus

- Un sous-processus est donc un processus à part entière. En termes de notation, son libellé est noté en dessous du rectangle. De plus, contrairement à celui d'une tâche exprimé par un verbe, le libellé d'un sous processus est un nom, généralement le substantif d'un verbe.

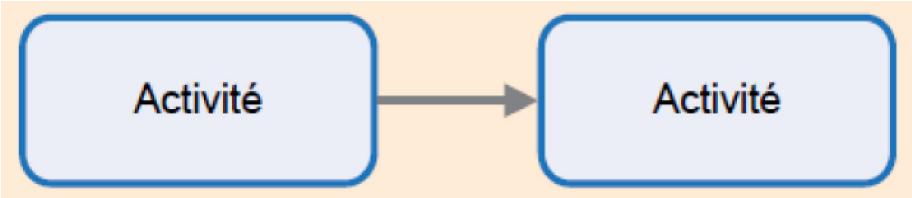


Concepts de base
Activités – Sous-processus

- Par exemple, l'activité « préparer commande » devient le sous-processus « préparation de la commande ».

Concepts de base
Flux séquentiels

Utilisés pour montrer la progression du flux.



- Tâches,
- Sous processus.

Concepts de base

Événements

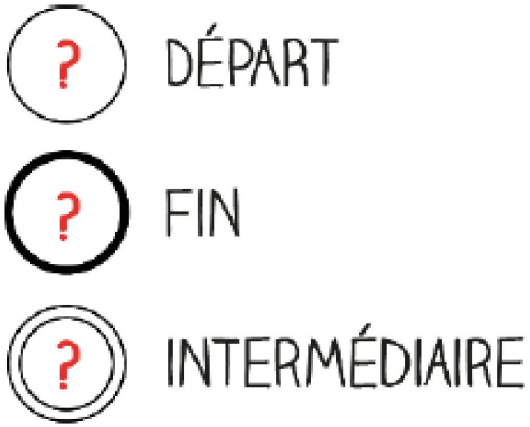
Utilisés pour débuter ou finir un processus et gérer des actions spécifiques au cours de celui-là.

Dans un modèle de processus, il existe trois catégories d’ évènement :

- Evènements de départ,
- Evènements de fin,
- Evènements intermédiaires.

Concepts de base
Événements

Un des principes de base de BPMN est de pouvoir ajouter des **pictogrammes** aux différents concepts, afin de spécifier leur nature.
Sur un évènement, ces icones s'inscrivent à **l'intérieur du cercle**.
Notons qu'il **n'est pas obligatoire** de préciser la nature de l'évènement et que le cercle peut rester vide.



Concepts de base
Événements de départ

Ces évènements déclenchent le processus.

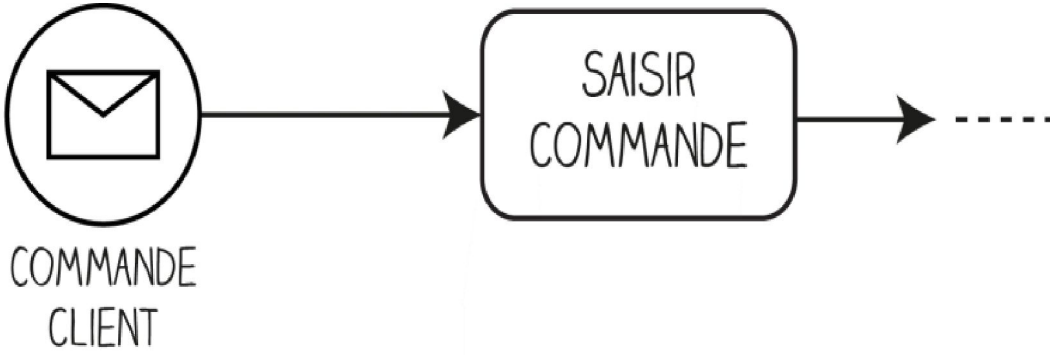
Ce sont des évènements de type « ***catch*** » c'est-à-dire des évènements dont notre processus est le destinataire ou l'attrapeur.

Concepts de base
Événements de départ

La star des évènements, c'est le ***message***, représenté par une enveloppe.

Le processus démarre suite à la réception d'un message provenant de l'extérieur du processus, peu importe sa forme (oral, courrier, email, etc…).

Par exemple, la saisie de la commande démarre lorsque le client donne sa commande.



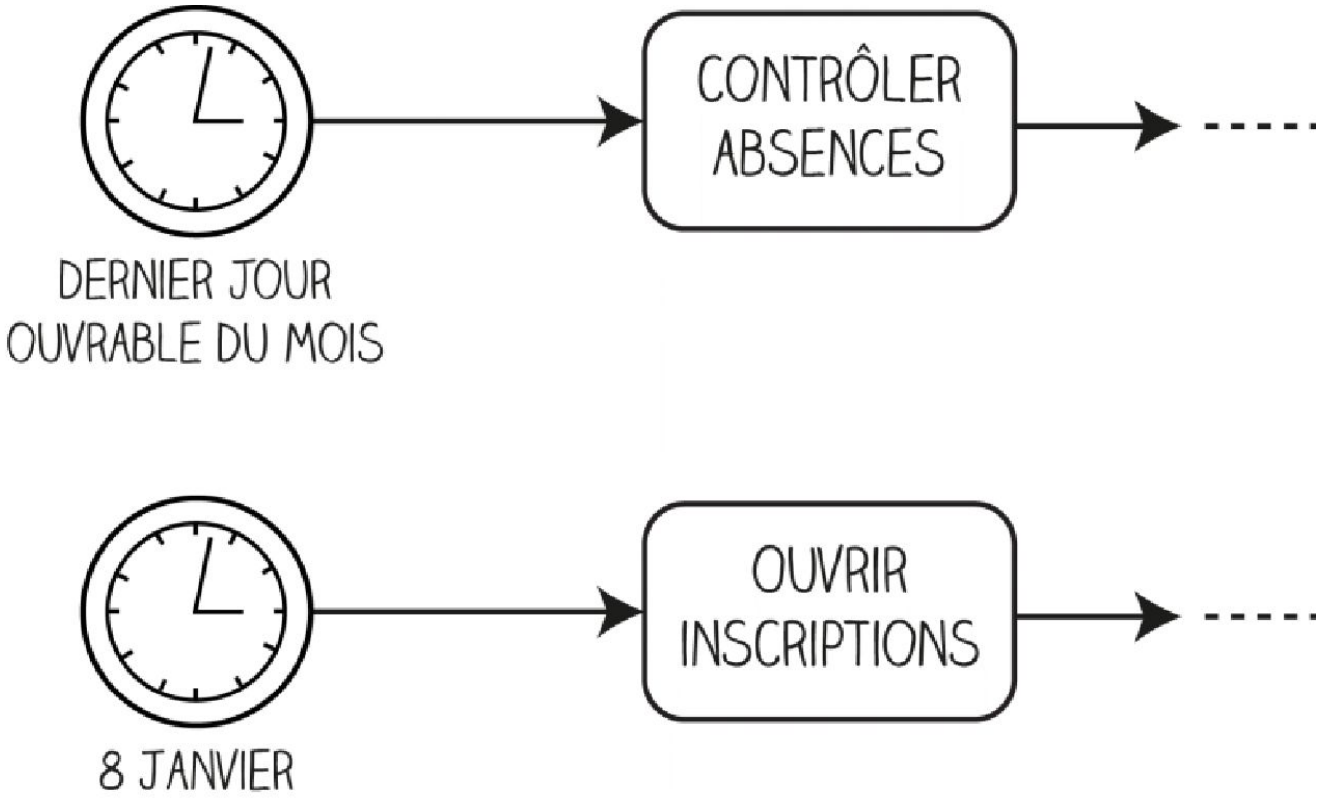
Concepts de base
Événements de départ

Autre tête d'affiche, le ***timer***, dont le pictogramme est une horloge.

Cet évènement correspond à une indication temporelle comme une date, une heure ou une périodicité. Le processus démarre lorsque la condition temporelle est vérifiée.

Concepts de base
Événements de départ

Par exemple, la réalisation du processus de paye démarre le dernier jour ouvrable du mois ou encore l'ouverture des inscriptions démarre le 8 janvier.



Concepts de base
Événements de départ

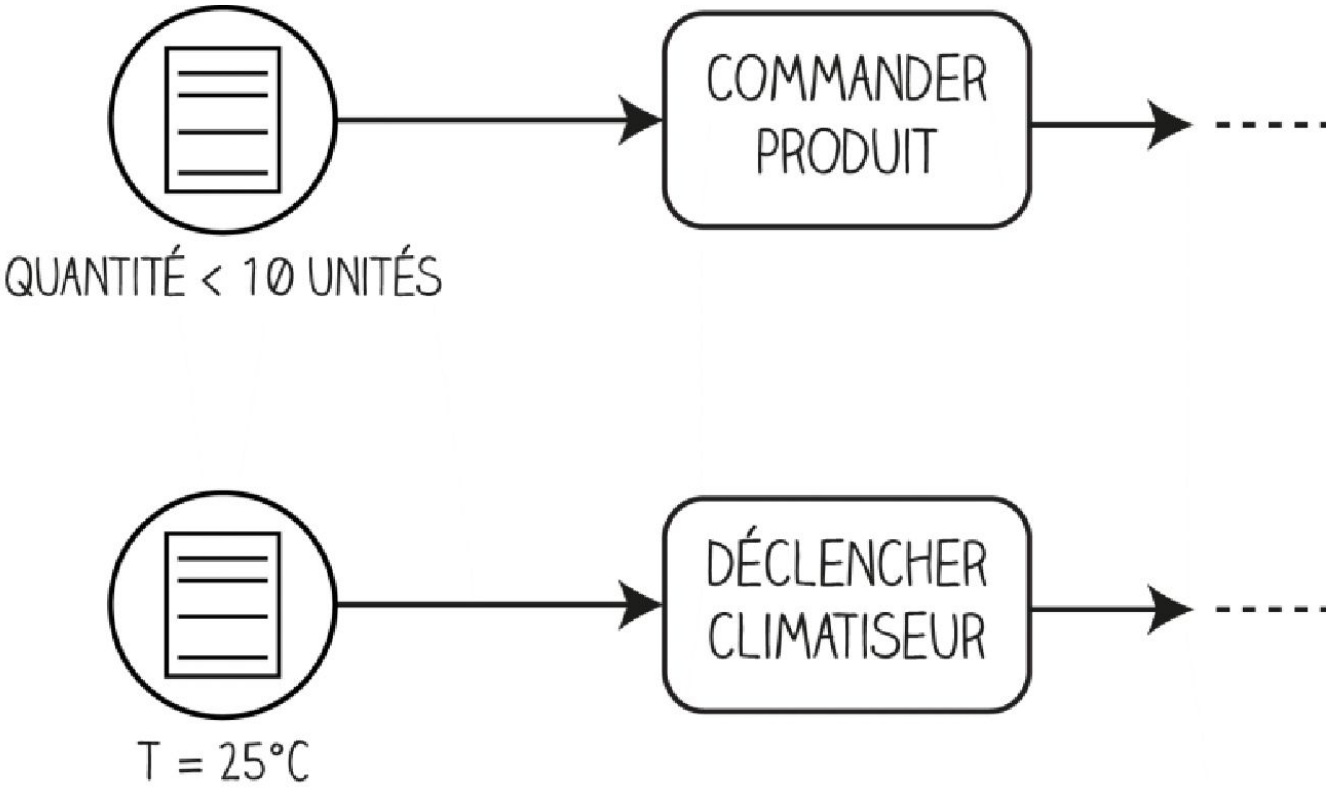
La **condition** peut également porter sur une donnée.

Le type d'évènement conditionnel permet de formaliser le déclenchement d'un processus suivant une **règle de gestion**.

Concepts de base

Événements de départ

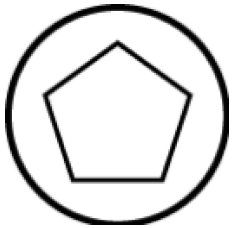
Par exemple, le processus de réapprovisionnement démarre lorsque la quantité du produit est inférieure à 10 unités ou encore lorsque la température atteint 25° cela déclenche le processus de régulation de la température.



Concepts de base
Événements de départ

Lorsque plusieurs évènements déclenchent le processus, on utilise alors le type **multiple**.
On spécifie dans le libellé les différents évènements.
Deux formalismes existent correspondant à deux modes de déclenchement différents.

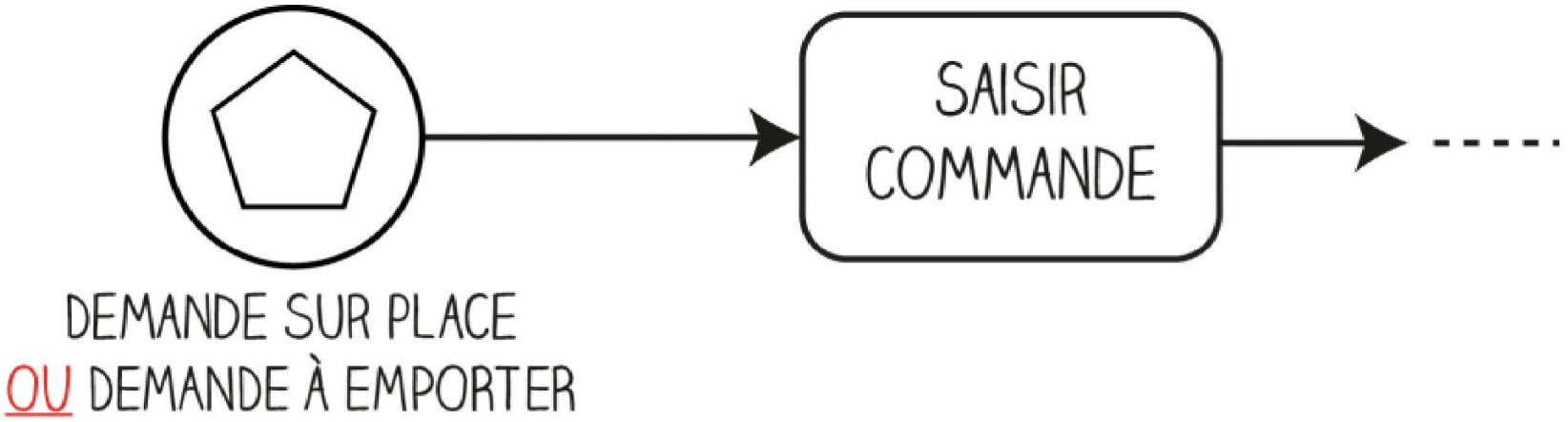
Lorsque seule l'occurrence d'un des évènements démarre le processus, on utilise un pentagone.



Lorsque l'occurrence de tous les évènements sont nécessaires au déclenchement du processus, on appelle cela un multiple-parallèle et le signe utilisé est un + .

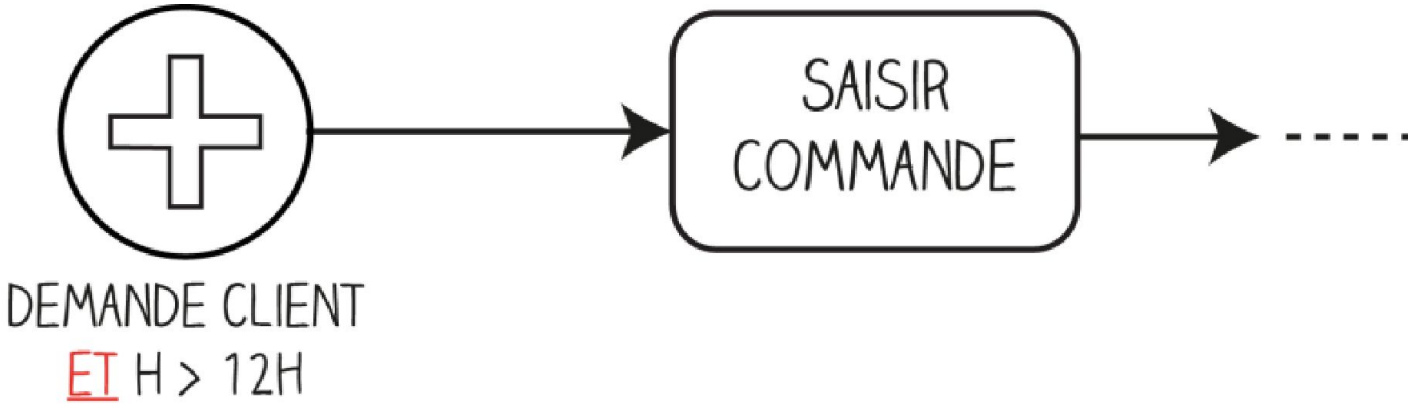
Concepts de base
Événements de départ

Par exemple, la saisie de la commande peut démarrer pour une demande de plat sur place ou une demande de plat à emporter.



Concepts de base
Événements de départ

Autre contexte, la saisie de la commande pourrait démarrer par la demande du client mais également s'il est midi passé. Il faut que tous ces évènements soient réunis pour passer commande.



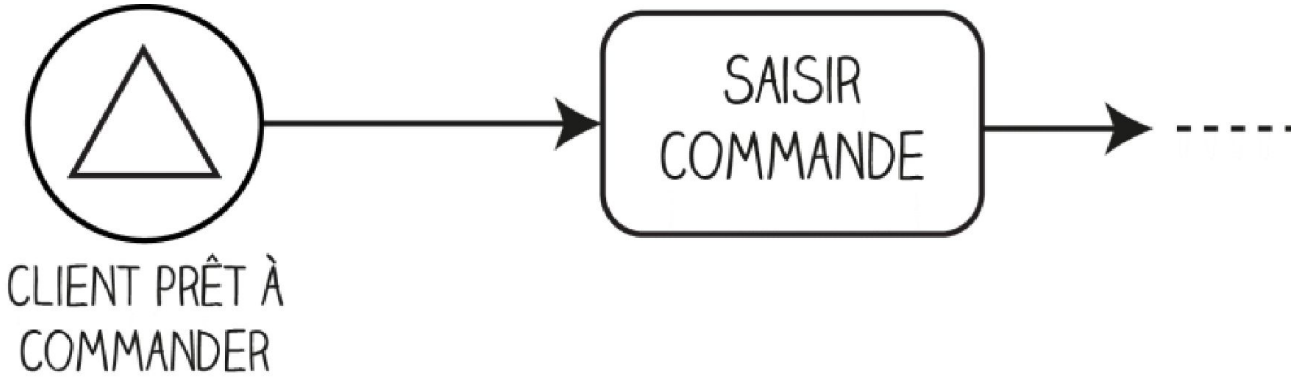
Concepts de base
Événements de départ

Le **signal**, représenté par un triangle.

Le processus se déclenche après avoir capté un signal d'un autre processus.

Concepts de base
Événements de départ

Par exemple, notre processus de service client dans un restaurant pourrait commencer par le signal du client qui ferme sa carte, lève la main ou scrute la salle à la recherche d'un serveur disponible.
Ce n'est pas un message destiné à un serveur en particulier mais un signal qui sera capté par un des serveurs et déclenchera une instanciation du processus.



Concepts de base
Événements de fin

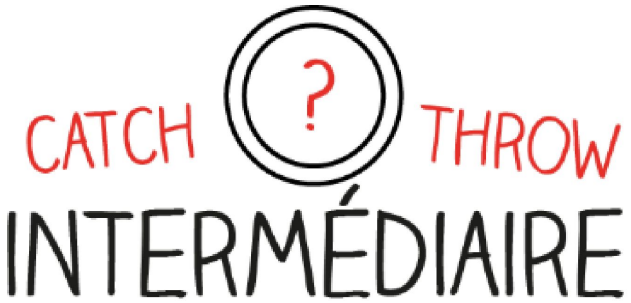
Ce sont des évènements de type « ***throw*** » c'est-à-dire dont notre processus est l'émetteur ou le déclencheur.

Nous retrouvons donc le **message** et le **signal** en mode throw, c'est-à-dire destiné à un autre processus. Il existe également l'évènement **multiple** mais peu utilisé.

Nous présentons en plus l'évènement de terminaison ou **terminate**, formalisé avec un remplissage du cercle.

Concepts de base
Événements intermédiaires

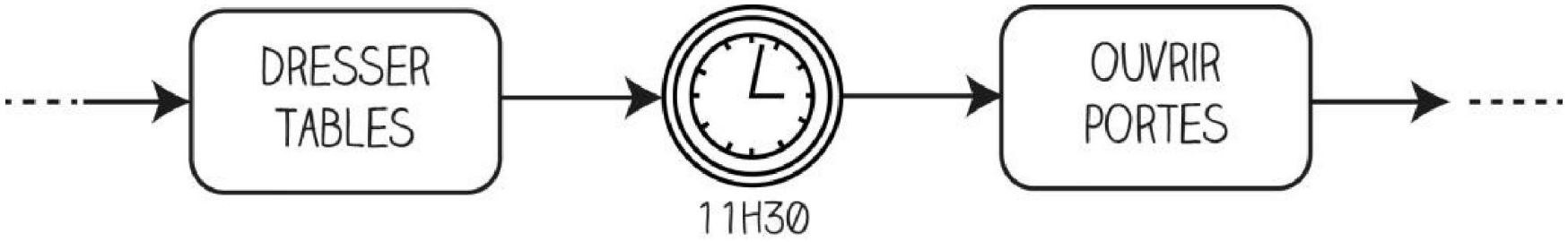
Les évènements intermédiaires se produisent eux, au cours d'un processus. Ils peuvent être de type « ***catch*** » ou « ***throw*** ».
Le formalisme des pictogrammes va donc devoir spécifier pour certaines natures d'évènement si notre processus est le déclencheur ou le destinataire.



Concepts de base
Événements intermédiaires

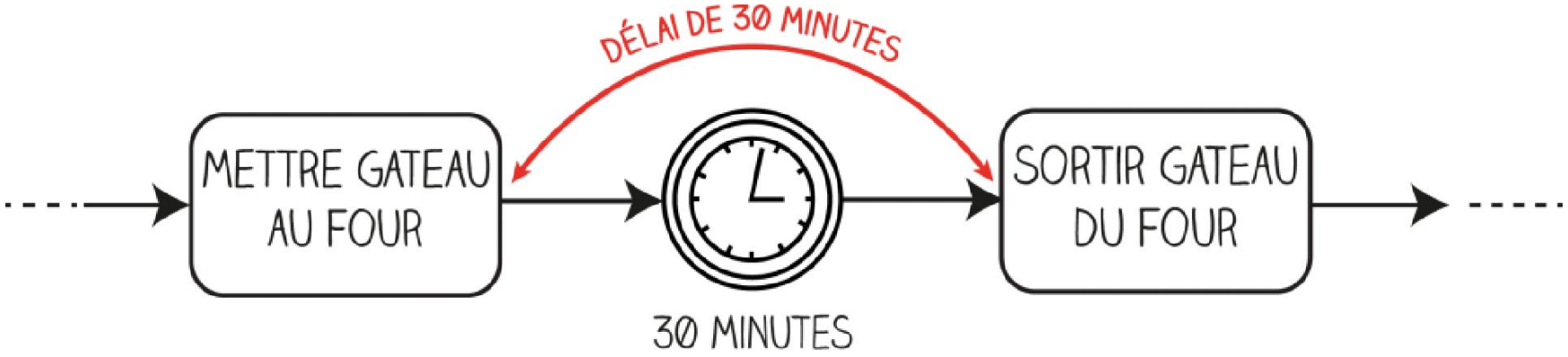
Nous retrouvons tout d'abord le ***timer***, l'un des plus utilisés qui permet de modéliser un délai d'attente ou une échéance entre deux activités.

Par exemple, dans le processus de préparation de la salle de restauration, il faut attendre 11H30 avant d'ouvrir les portes du restaurant, peu importe si le dressage des tables a été fini avant.



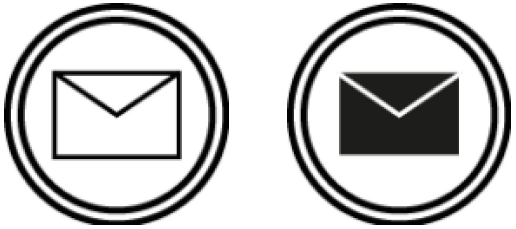
Concepts de base
Événements intermédiaires

Ou encore, il faut attendre 30 minute entre la mise au four et l'enlèvement du gâteau et encore 2H avant de le déguster. Notez que le délai se calcule à partir de la fin de l'activité jusqu'au début de la suivante.



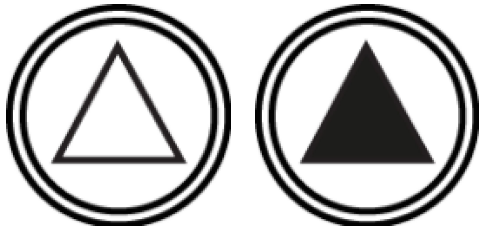
Concepts de base
Événements intermédiaires

Pour les évènements intermédiaires de types **message**, on distingue l'envoi ou la réception par le remplissage de l'enveloppe.



Exemple : le client peut demander la note et notre serveur la lui remettre une fois éditée. Attention, il faut toujours se placer au niveau du processus étudié. La demande de note est effectivement envoyée par le client. Mais le processus représenté ici, est celui du serveur, qui lui reçoit le message. C'e

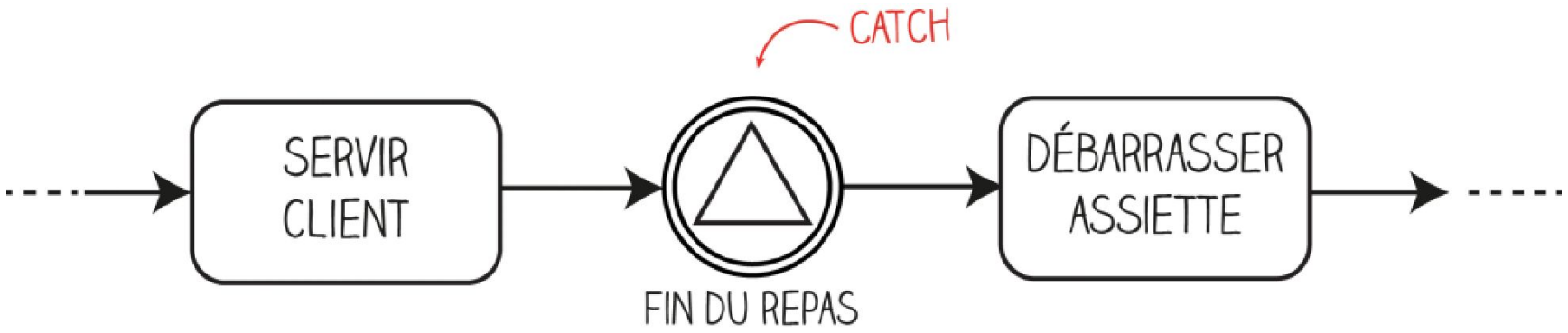
Concepts de base
Événements intermédiaires



Il en est de même pour les évènements intermédiaires de type **signal**.
Nous précisons que pour qu'un signal émis face référence à un signal reçu, ils doivent être libellés de la même façon.
On les utilise pour deux raisons : pour broadcaster un message à plusieurs destinataires, définis ou non mais surtout pour s'affranchir de la contrainte qu'un message ne peux pas être envoyé dans un même processus.
En effet, la norme BPMN n'autorise pas l'envoi-réception de message au sein d'un même processus.

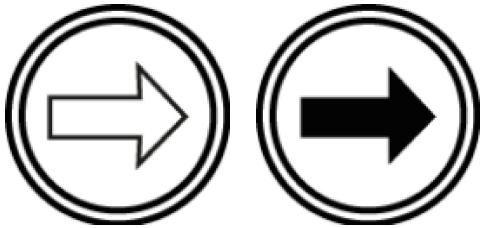
Concepts de base
Événements intermédiaires

Dans notre exemple, une fois son plat terminé, le client peut émettre un signal au serveur qui viendra le débarrasser.



Concepts de base
Événements intermédiaires

L'évènement de type **Lien** est plus un élément d'aide graphique qu'un véritable évènement. En effet, la modélisation complète de certain processus peut vite devenir imposante et donc perdre en lisibilité. Le lien permet de faire un renvoi entre deux parties de processus. La séquence de flux est découpée pour une meilleure visibilité. Les évènements lien marchent par paires et il est obligatoire de donner le même nom aux deux liens se faisant référence.



Concepts de base
Événements intermédiaires

Par exemple, supposons que notre processus devienne vraiment trop grand et doivent être découpé.

Je termine la première partie par un évènement lien de type throw et je récupère le flux par un évènement lien de type catch.

C’est juste une question de représentation graphique.

Cela n’a aucune incidence sur le processus.

Nous venons de voir les concepts de base afin de réaliser nos premiers modèles de processus.

Concepts de base
Événements intermédiaires