

Utilisation d'UML à des fins de recueil des besoins et d'analyse 2/2.



A l'issue de la formalisation des besoins

- La totalité des cas d'utilisation
 - Leurs relations
 - Les acteurs concernés
- Les scénarios de chaque cas d'utilisation
- Les diagrammes d'interactions

Les objectifs de l'analyse

- Formalisation des besoins utilisateurs
- Identification des principaux artefacts de l'application (analyse proprement dite)

Les diagrammes utilisés

- Les diagrammes d'activité
- Les diagrammes de séquence
- Les diagrammes de classe « BEC »

Que sont ces artefacts?

- Il s'agit d'une première estimation des classes qui réaliseront l'application.
- Elles se divisent en 3 catégories :
 - Les classes d'interface (ou boundary)
 - Les classes contrôleur
 - Les classes entités.
- Chaque cas d'utilisation doit pouvoir être réalisé par la collaboration de ces 3 types de classes.

Les classes Boundary

- Elles représentent l'interface entre l'acteur d'un cas d'utilisation et le système.
- Il peut s'agir de :
 - Fenêtre « Windows »
 - Pages HTML
 - D'interfaces vers des dispositifs électro-mécaniques.
 - De protocoles d'échange avec d'autres systèmes :
 - Tuxedo
 - CICS
 - SOAP
 - Web services
 - ...

Les classes contrôleur

- Elles représentent le processus, c'est à dire la gestion des interactions identifiées lors de la formulation du recueil des besoins et de l'analyse.
- Ces classes dirigent les activités entre les classes boundary et les classes métier.

Les classes métier

- Ce sont les classes décrites dans les cas d'utilisation qui représentent les données manipulées lors des processus.
- Bien souvent ces classes correspondent à des informations persistantes :
 - Elles sont sauvegardées sur un support permanent tel qu'un SGBD

Les stéréotypes

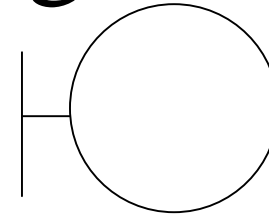
- Définition :
 - Une classe d'élément de modélisation.
- Un stéréotype est utilisé pour définir une utilisation particulière d'un élément de modélisation.
- Un stéréotype permet la méta-classification d'un élément de modélisation.
- Un stéréotype peut être associé à un pictogramme particulier qui le rend aisément identifiable.
- Il est autrement représenté par son nom entre << >>

Exemples de stéréotypes de classe

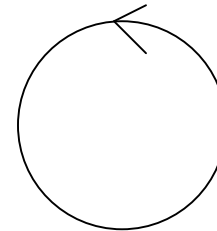
- Un acteur <<Actor>>
- Une classe boundary <<Boundary>>
- Une classe entité <<Entity>>
- Une classe contrôleur <<Controler>>
- Design pattern <<Pattern>>
- ...

Les stéréotypes des classes BEC associés à leur pictogramme

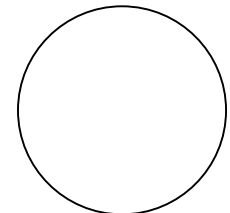
Les classes boundary



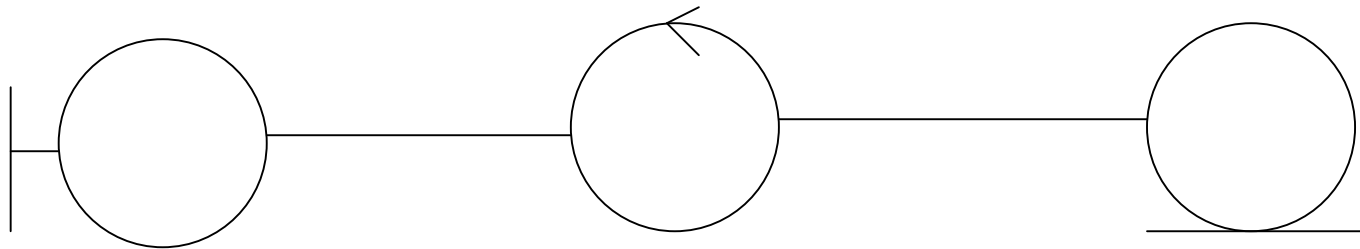
Les classes contr leur



Les classes entit s



Expression du modèle BEC



Tout use case doit pouvoir être réalisé par l'association d'au moins une classe boundary, exactement une classe contrôleur, au moins une classe entité .

Les contraintes

- Les acteurs n'interagissent qu'avec les classes « boundary »
- Les classes « boundary » n'interagissent qu'avec les classes « controleur »
- Les classes entité n'interagissent qu'avec des classes « controleur »
- Les classes « controleur » interagissent avec tous les classes, y compris avec des classes « controleur »

Comment identifier ces classes

- Les classes boundary sont la réalisation des échanges identifiés pendant la formalisation du recueil des besoins.
- Les classes « contrôleurs » sont déduites des cas d'utilisation.
- Les classes métier sont une abstraction des informations manipulées. Une analyse des différents substantifs utilisés peut s'avérer précieuse.

Exemples de classes

- Boundary :
 - Formulaire de log-in
 - Formulaire de réservation d'ouvrage
 - Formulaire de refus de réservation
 - Formulaire de confirmation de réservation
 - Menu principal
- Contrôleur
 - Contrôleur de log-in
 - Contrôleur de réservatin
- Métier
 - Informations élève (FicheElève)
 - Ouvrage (FicheLivre)

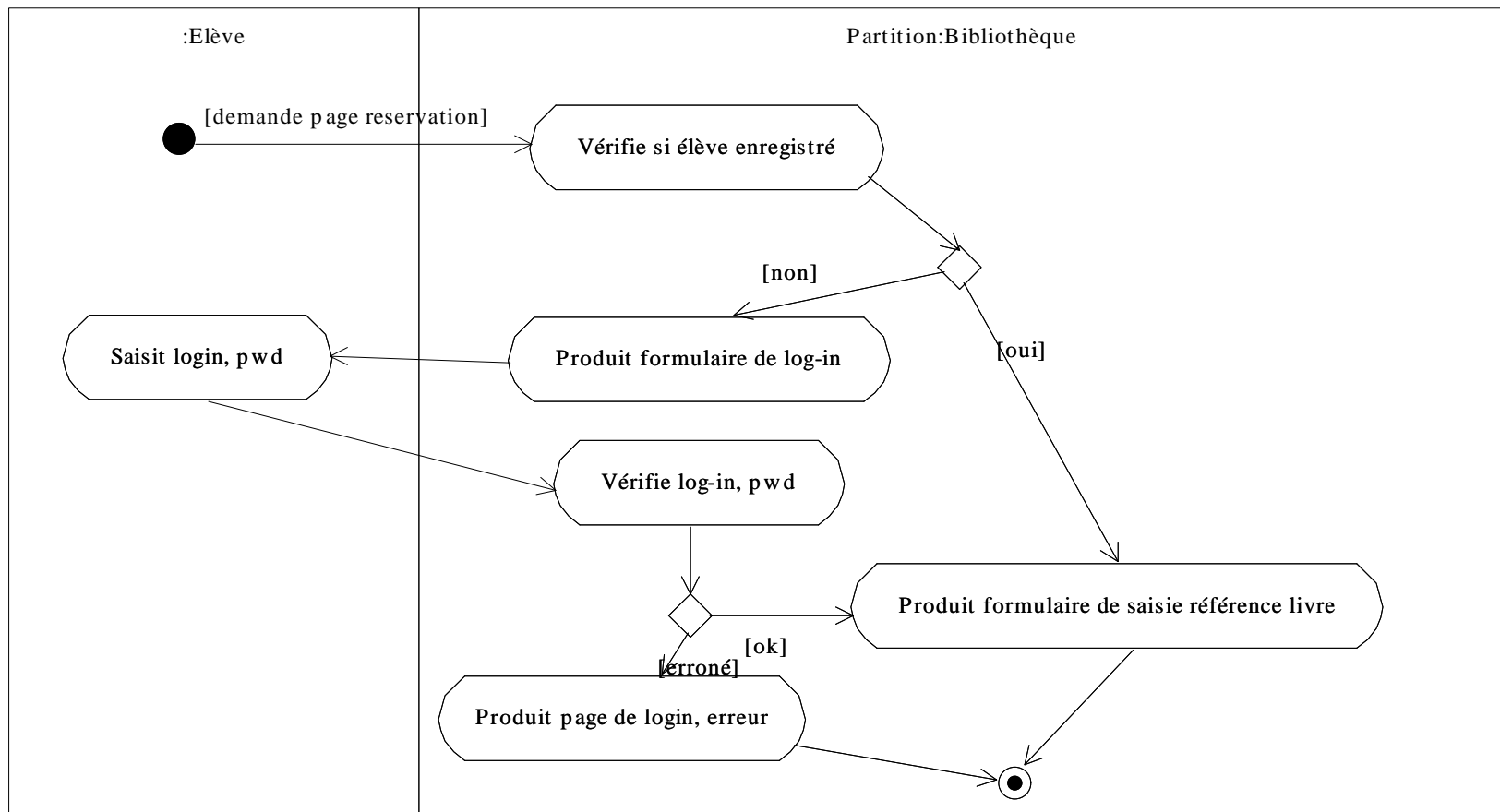
Les diagrammes d'activité

- Un diagramme d'activité représente l'état de l'exécution d'un mécanisme sous forme d'un déroulement d'étapes regroupées séquentiellement dans des branches parallèles de flot de contrôle.
- Il peut exister, optionnellement, un état initial et un état final.

Utilisation dans la cadre de l'analyse

- Ils servent essentiellement à décrire les activités de l'application lors de la réception de messages émis par un acteur.
- Ils sont associés à un ou plusieurs diagrammes de séquence.

Exemple de diagramme d'activités

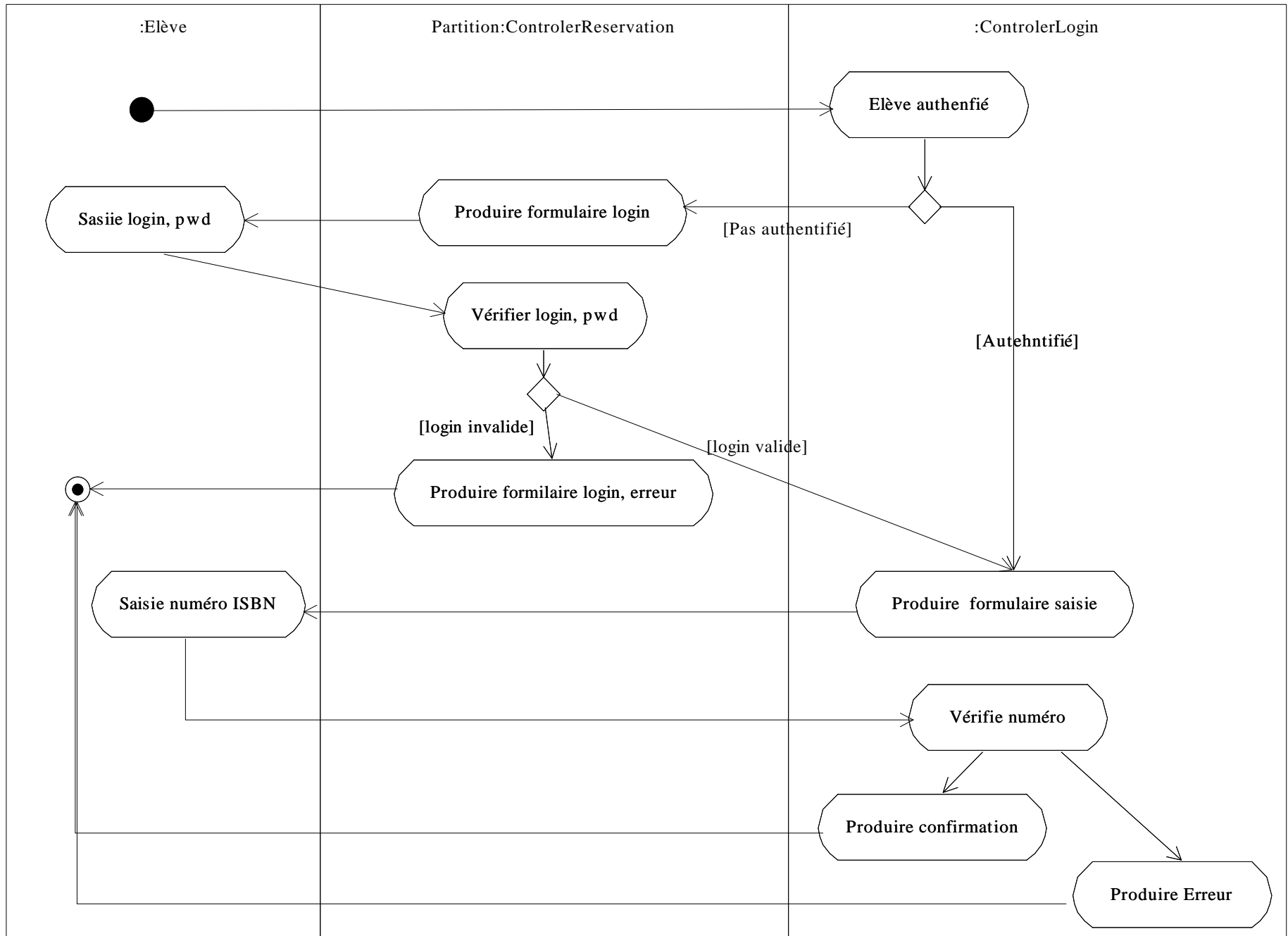


Les éléments des diagrammes d'activité

- Les travées
- Les états
- Les transitions
- Les conditions
- Les synchronisations
- Les signaux

Les travées

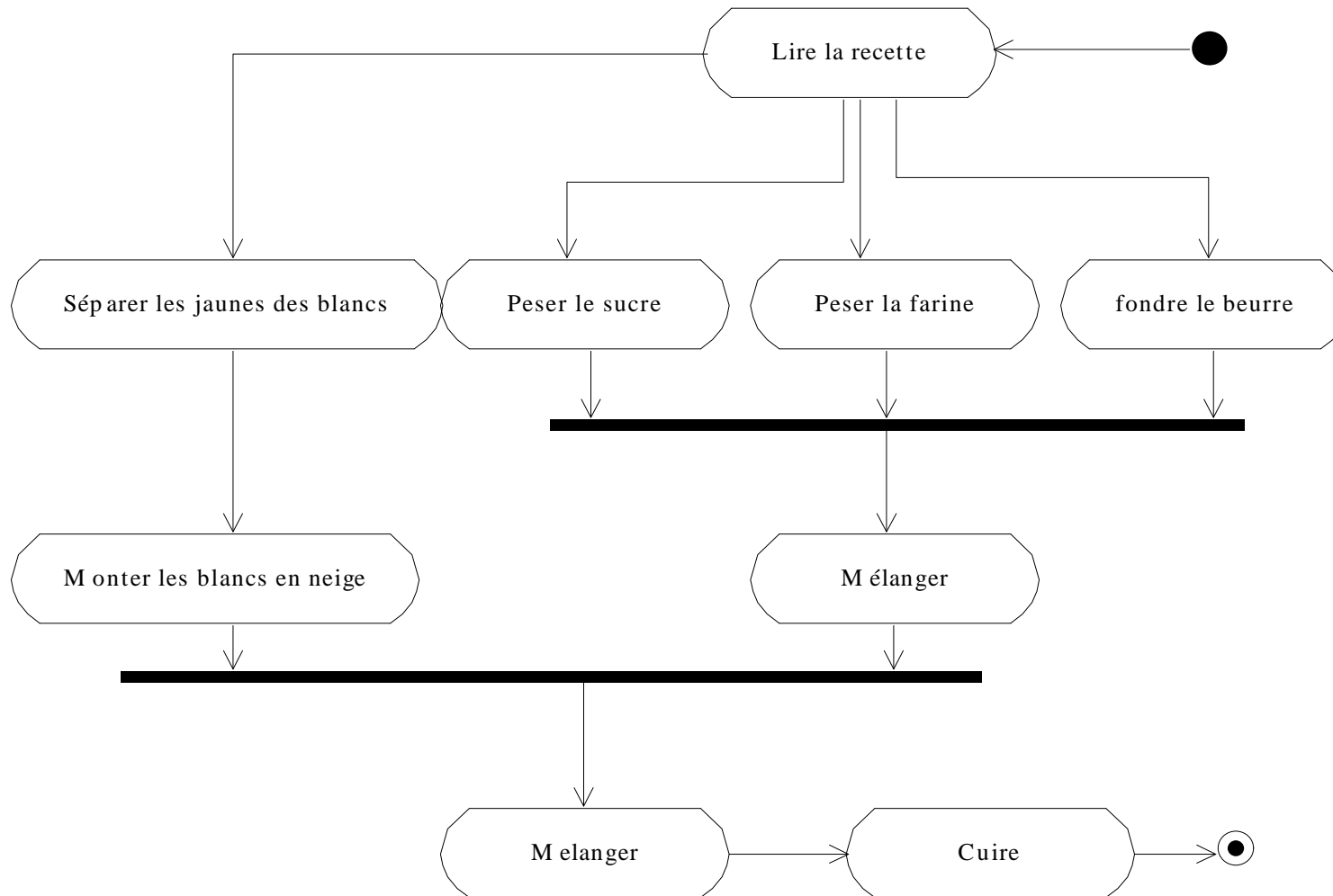
- Ce sont des partitions qui permettent de mettre en exergue les différentes responsabilités au sein d'un mécanisme



Les synchronisations

- Ce sont les moyens par lesquels une travée peut ouvrir ou fermer des branches parallèles au sein d'un flot d'exécution.
- Une synchronisation ne peut être franchie que lorsque toutes les transitions entrantes sont déclenchées.

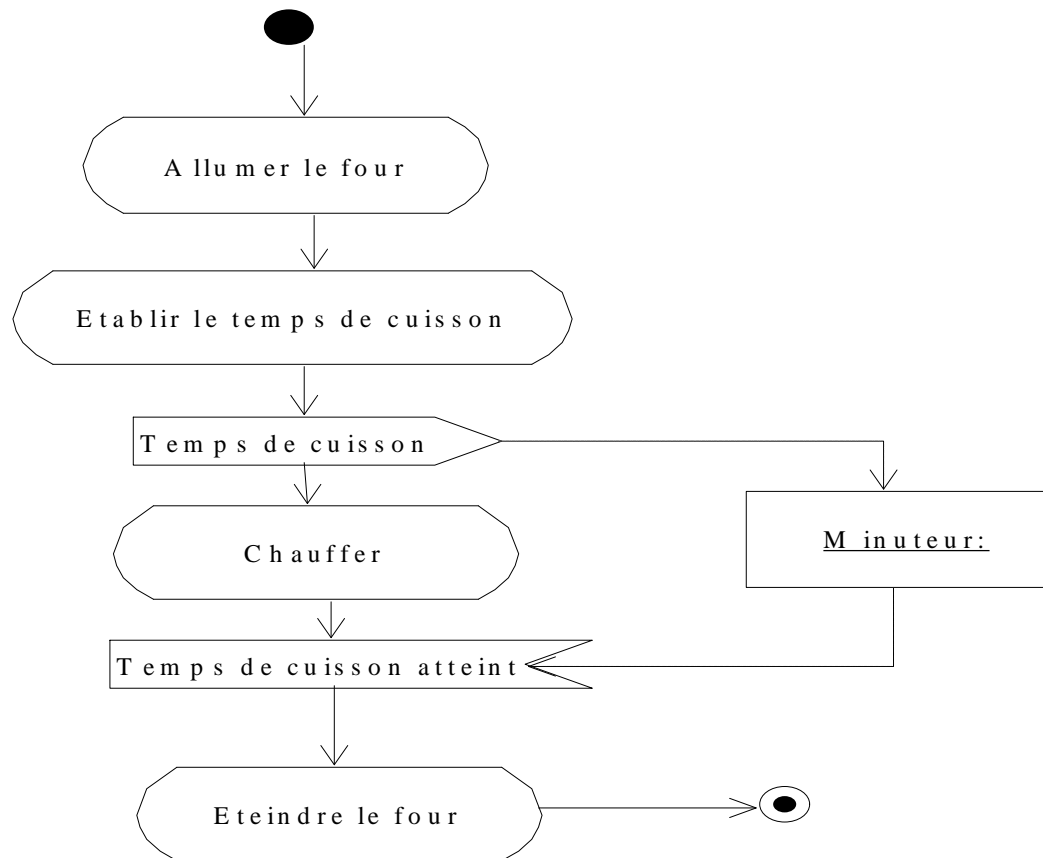
Exemple de synchronisations



Les signaux.

- Ce sont des stéréotypes qui représentent graphiquement l'envoi et la réception de signaux.
- Ils permettent de représenter explicitement des informations relatives aux transitions.
- Ils peuvent être reliés à l'objet émetteur du signal

Exemple de signal



Analyse des use case

- Réaliser un diagramme d'activité simple par use case.
- En déduire :
 - La nature des collaborations avec d'autre use case
 - Les règles métier, algorithmes, contraintes légales applicables.

Analyse des use case

- Raffiner le diagramme d'activités en répartissant les activités entre les différents contrôleurs impliqués dans un use case.
- Ne pas faire intervenir les classes boundary, ni les classes métier.

Analyse des use case.

- Réaliser un diagramme de séquence pour tous les scénarios de use case importants :
 - Qui correspondent à un grand nombre d'interactions.
 - Qui impliquent plusieurs use case

Analyse des use case

- Pour chaque classe utilisée dans un use case, les éléments suivants sont mis en évidence :
 - Ses attributs
 - Ses méthodes
 - Eventuellement ses collaborations.

Analyse des use case

- La même classe pouvant être utilisée dans des scénarios différents d'un même cas d'utilisation, dans des cas d'utilisation différents :
 - Ses attributs et ses rôles sont consolidés au fur et à mesure de l'analyse.
 - Les incompatibilités sont détectées et traitées le cas échéant.

Résultats de l'analyse

- Les classes boundary, entité, contrôleur sont munies de toutes les méthodes et tous les attributs nécessaires à la réalisation de tous les uses case et de tous les scénarios.