

UML

2 - Diagramme de séquence « système »

Diagramme d'activité

Diagramme d'états-transitions

Diagrammes de comportement

Analyse fonctionnelle

Bertrand LIAUDET

SOMMAIRE

SOMMAIRE	1
LE DIAGRAMME D'ACTIVITE	3
1. Diagramme d'activité et analyse fonctionnelle (20 slides, de 1 à 20)	3
Contexte (slide 1)	3
Complément sur les UC : scénario, scénario nominal (slide 2)	4
Présentation du diagramme d'activité (slide 3)	5
Usages (slide 7)	9
2. Syntaxe des diagrammes d'activité - vocabulaire et graphique	11
Éléments de base : Activité – Action – Transition – Début – Fin (slide 9 et 10)	11
Les transitions (slide 11)	13
Élément particulier : action à sous-activité (slide 19)	21
Élément particulier : diagramme d'activité à travées (slide 20)	22
3. Exercices	24
3-1 : Le save as de Word	24
3-2 : Recette de cuisine : diagramme d'activité	25
3-3 : Arrivée d'un nouvel employé : diagramme d'activité à travées	26
3-4 : Diagramme d'activité des scénarios d'un UC	27
LE DIAGRAMME DE SEQUENCE « SYSTEME »	28
1. Le diagramme de séquence système (11 slides, de 21 à 31)	28
Contexte (slide 21)	28
Les bases (slide 22)	29
Diagramme de séquence système avec des acteurs passifs (slide 26)	33
Diagramme de séquence système avec des sous-systèmes (slide 28)	35
Variante à éviter : Description textuelle des UC	37
2. Variantes à éviter : Modularité, test, boucle, autres contraintes	38
Modularité : à éviter	38
Test, boucle, option : à éviter	40
Autres contraintes, commentaires et notes	44
3. Utilisation en architecture Système et Réseaux	45

Communication entre les serveurs (slide 30)	45
Exemple : protocole d'authentification Kerberos (slide 31)	46
4. Exercices	47
4-1 : DSS du SNUC	47
4-2 : Protocole d'identification Kerberos	48
DIAGRAMME D'ETATS-TRANSITIONS	49
1. Présentation (11 slides, de 32 à 42)	49
Contexte (slide 32)	49
Principes (slide 33)	50
Représentation UML	50
Exemple (slide 34)	51
2. Syntaxe des diagrammes d'états-transitions - vocabulaire et graphique	53
État (slide 36)	53
Transition (slide 38)	55
3. État composite (slide 39)	56
4. Automate d'états finis (slide 41)	58
5. Exercices	60
5-1 : La porte de garage	60
5-2 : La montre digitale	61

Edition juin 2025

LE DIAGRAMME D'ACTIVITE

*Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.
La méthode est dénuée de sens tant qu'elle est déconnectée du rapport au savoir.*

1. Diagramme d'activité et analyse fonctionnelle (20 slides, de 1 à 20)

Contexte (slide 1)

- Les diagrammes d'activité peuvent être utilisés en analyse fonctionnelle et en analyse technique. Toutefois, ils sont surtout utilisés en analyse fonctionnelle.
- Il y a 4 diagrammes UML qu'on utilise en analyse fonctionnelle :

	Diagramme UML
ANALYSE FONCTIONNELLE	Cas d'utilisation - UC
	Activité
	Séquence
	États-transitions

- On ne traite dans ce chapitre que du diagramme d'activité.

Exemple

- Il y aura **plusieurs scénarios pour l'UC « retirer de l'argent »** du GAB : si le code de la carte de retrait est faux ; si le client n'est pas autorisé à retirer de l'argent ; si le guichet n'a plus de billets ; etc.
- **Le scénario nominal décrit le retrait d'argent « normal »**, c'est-à-dire ce pour quoi est fait l'UC. Dans un scénario nominal, il n'y a pas d'alternatives.

Définitions

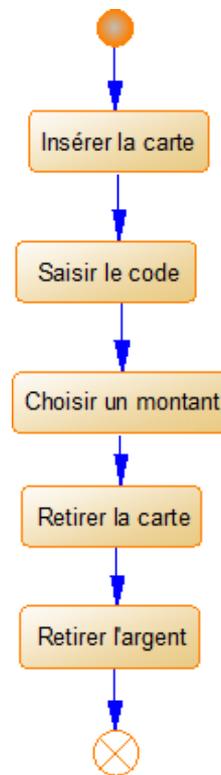
- Un UC est un ensemble de chemin possibles qui sont prenables dans la réalisation d'une activité. Par exemple, quand on retire de l'argent, il y a plusieurs alternatives : erreur de code, montant trop élevé, etc.
- **Un scénario est un déroulement concret d'un UC** parmi tous les déroulements concrets possibles (autrement dit toutes les alternatives).

⇒ Un UC est un ensemble de scénarios.

⇒ Le **scénario nominal** est le scénario pour lequel le UC est conçu.

Exemple de base : retirer de l'argent au GAB

- On présente la **succession des actions**.
- On a **un point de départ** et un seul.
- On a un (ou plusieurs) **point d'arrivée**.
- Ici traite le scénario nominal de l'UC :
 - ⇒ Il n'y a pas les alternatives.
 - ⇒ **Les transitions sont simples** : ce sont les flèches.
- Pour améliorer la lisibilité, **on représente le diagramme plutôt verticalement**. Ça prend de la place, mais c'est finalement plus clair.



Définition : vocabulaire (slide 4)

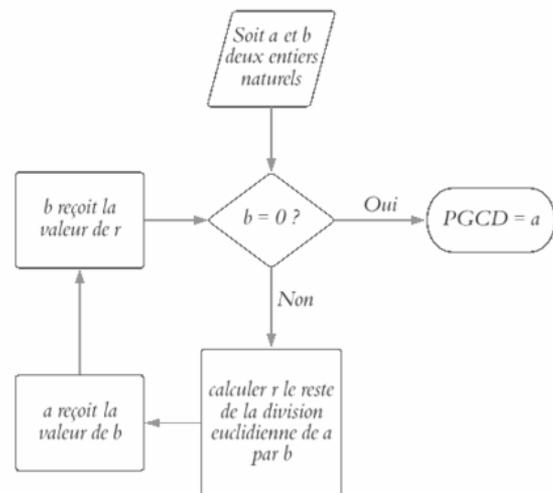
Une **activité** est une succession d'**actions**.

Principes

- Le diagramme d'activité est un **diagramme de comportement**.
- Il s'applique à tout système, que ce soit un programme ou une entreprise.
- Il permet de présenter une activité du système avec :
 - **une succession d'actions** qui se déroulent dans le temps,
 - **un début**,
 - **une ou plusieurs fins**,
 - **des transitions entre les actions** qui pourront, entre autre, porter **des alternatives**.
- Il permet donc de **présenter un algorithme**, quel qu'il soit (algorithme = suite d'opérations pour résoudre un problème).
- C'est un diagramme facile à construire.
- La notion de « transition » est le lieu des subtilités.

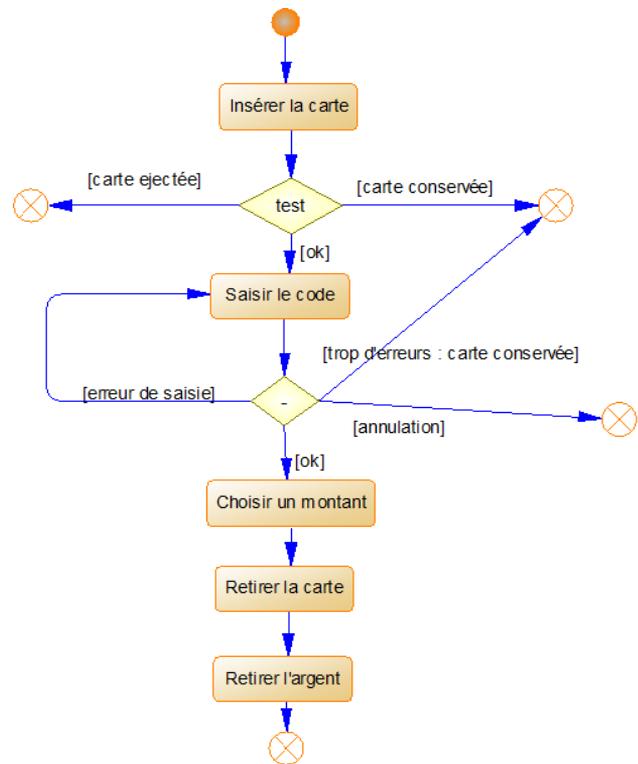
Organigrammes de l'algorithmique classique (slide 5)

- Le diagramme d'activité correspond aux « organigramme » utilisés dans l'algorithmique des années 60.
⇒ Attention, ce n'est pas l'organigramme d'une entreprise au sens de l'organisation hiérarchique et par service.
- Ces organigrammes étaient des algorithmes
- Une action c'est n'importe quel bloc d'instructions.
- Les tests sont des transitions particulières sont représentés par des losanges.
- Les boucles sont représentées par un retour en arrière dans le flot des actions.



Exemple avec test et boucle : retirer de l'argent au GAB (slide 6)

- On traite les cas particuliers des 2 premières actions : « insérer la carte » et « saisir le code ».
- On a deux transitions à décision (les losanges entre 2 flèches).
- On a 4 alternatives de sortie et une alternative de boucle.

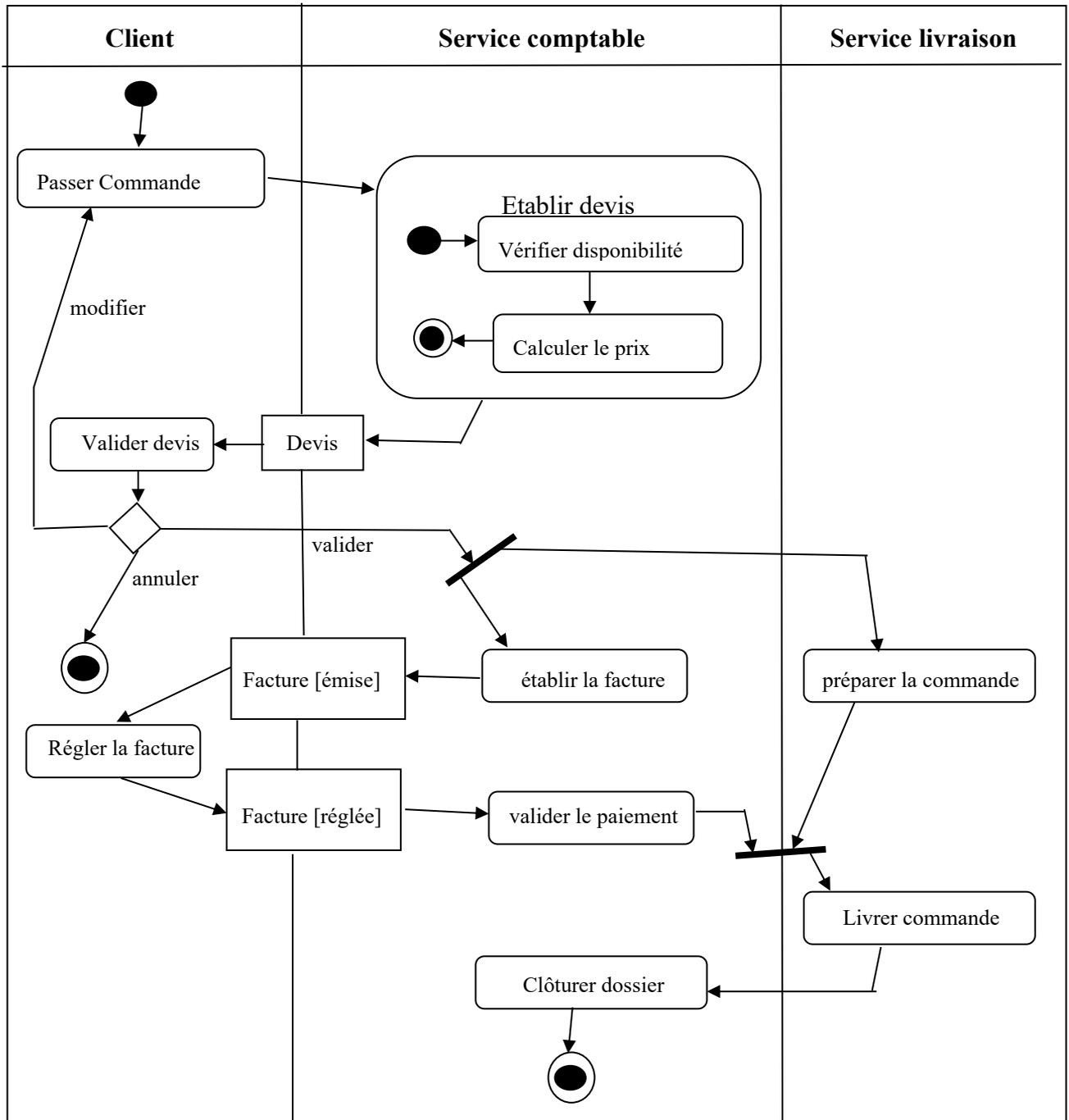


Usages (slide 7)

- Le diagramme d'activité est principalement utilisé au niveau de l'analyse fonctionnelle
- Il peut s'utiliser en analyse technique, mais c'est plus rare.
- On va l'utiliser dans l'analyse fonctionnelle pour décrire le déroulement d'un cas d'utilisation avec tous ses scénarios alternatifs en plus du scénario nominal.
- Ou peut aussi l'utiliser pour **décrire les interactions entre acteurs** du système. Ce sont les « **diagrammes à travées** » qui permettent ça.
⇒ C'est une approche qu'on retrouvait dans la méthode MERISE.

Diagramme à travée du traitement « PasserCommande » avec 3 acteurs (slide 8)

3 travées qui correspondent à 3 acteurs et potentiellement 3 postes de travail donc potentiellement 3 accès et usages possibles du système d'information.



● Etat initial

○ Etat final

Synchronisation —

action

transition →

données

« Établir devis » est une action à sous-activité.

2. Syntaxe des diagrammes d'activité - vocabulaire et graphique

Éléments de base : Activité – Action – Transition – Début – Fin (slide 9 et 10)

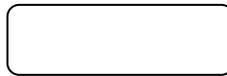
Activité (slide 9)

- Une activité est l'exécution d'un comportement, autrement dit un programme ou un bout de programme qui s'exécute.
 - Le diagramme d'activité décrit une activité : par exemple le « save as » ou le « passer commande ».
- ⇒ Un UC est une activité.

Action

Une action est une partie de l'activité.

Elle est représentée par une boîte aux angles arrondis (arrondi = traitement, droit = donnée).



Une activité est en général constituée de plusieurs actions qui se déroulent l'une après l'autre : elles seront **reliées par des transitions**.

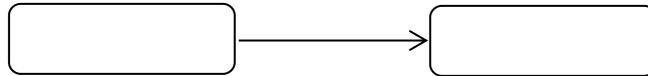
L'action permet de représenter n'importe quel bloc d'instructions, de la simple instruction au programme complet en passant par la fonction ou n'importe quel morceau de code.

Une activité peut être réduite à une action et une seule.

Transition (slide 10)

Une transition relie deux actions entre elles (action de départ --> action d'arrivée).

Elle est représentée par une flèche entre deux actions :



La transition se déclenche à la fin du comportement de l'action de départ.

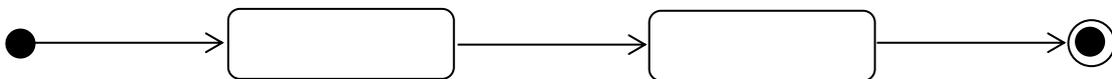
Etat initial

- L'état initial montre le point de départ de la première action. ●
- Il n'y a qu'un seul état initial par activité.

Etat final

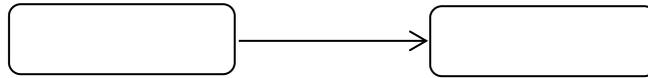
- L'état final montre le point d'arrivée de la dernière action. ⊙
- Il peut y avoir plusieurs états finaux pour une activité.

➤ *Exemple :*



La simple transition

Elle est représentée par une flèche entre deux actions :

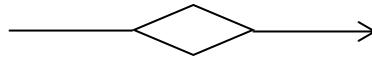


La transition se déclenche à la fin du comportement de l'action de départ.

Principe des autres transitions (slide 12)

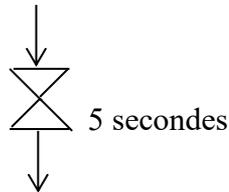
- On va mettre quelque chose au milieu de la simple transition pour préciser le fonctionnement de la transition

- Par exemple :



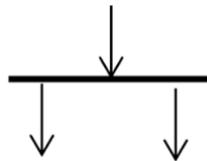
⇒ On a mis un « **losange d'alternative** » au milieu de la transition.

- Autre exemple :



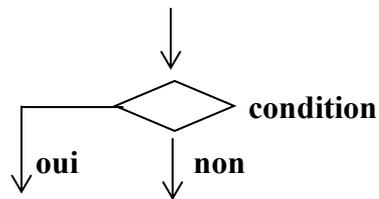
⇒ On a mis un « **sablier** » au milieu de la transition. En général il est commenté. Dans l'exemple, ça dit que la transition prend 5 secondes.

- Autre exemple :



⇒ On a mis un « **barre de synchronisation** » au milieu de la transition.

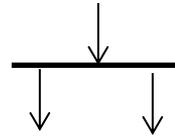
La transition conditionnelle = décision (slide 13)



- C'est un aiguillage dans une transition :
- On peut préciser la **condition** de l'aiguillage : ce n'est pas obligé.
- On précise toujours les conditions de passage (**oui** ou **non**) dans les différentes branches de l'aiguillage.
- **Attention** : la transition conditionnelle indique que le flux va partir d'un côté ou de l'autre. Mais attention : il n'y a pas d'actions (de saisie, d'affichage, d'affectation) à ce niveau. Le choix a été fait au niveau de l'action qui précède.

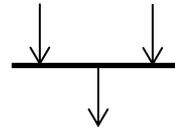
Transition fourche ou synchronisation fourche (slide 14)

- La transition fourche (barre horizontale) permet de faire démarrer 2 actions en parallèle à la suite d'une seule actions.



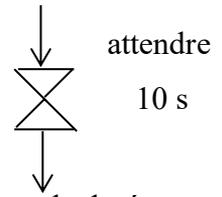
Transition jonction ou synchronisation jonction

- La transition jonction (barre horizontale) permet de faire démarrer 1 action après la fin de 2 autres actions.



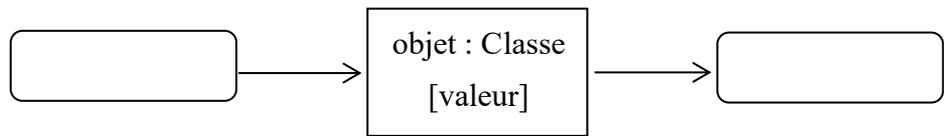
Transition à délai (slide 15)

- La transition à délai permet de modéliser un délai entre la fin d'une action et le début d'une autre.
- Ce délai peut dire qu'il faut attendre entre les 2 actions.
ou bien que l'action de départ doit durer le temps fixée.
- On peut aussi juste écrire « attendre » la fin d'un processus si on ne connaît pas la durée.



Transition à objet (slide 16)

- On peut faire apparaître les objets dans une transition entre deux actions.
- Les objets représentés sont en général les entrées ou les sorties des actions.
- Syntaxe UML :

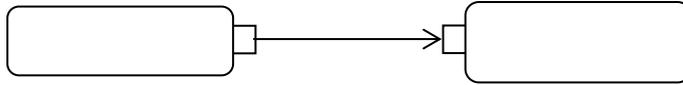


- Les données sont représentées par des rectangles.
- Le nom de la donnée est un objet (au sens de la POO) ou un simple attribut. On peut préciser sa classe (son type) et sa valeur.

Transition à « Pins » d'entrée et de sortie (slide 17)

- On peut faire apparaître les données (ou objets) au début ou à la fin d'une transition et donc en entrée ou en sortie d'une action.

- Syntaxe UML :



- Les petits carrés représentent les données : les « pins ».
- On peut avoir plusieurs pins différents en entrée et sortie d'une même action :



- On peut préciser le contenu des données des « pins ».

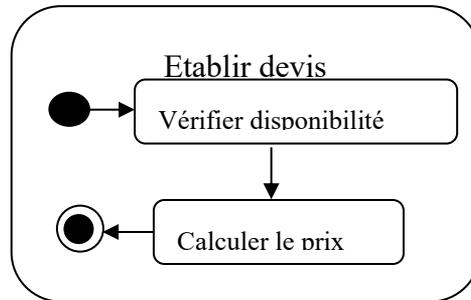
Transition à signal : envoi et réception (slide 18)



- On peut faire apparaître l'envoi ou la réception d'un signal entre deux actinos.
- Ca permet de modéliser l'enchaînement asynchrone d'actions.

Élément particulier : action à sous-activité (slide 19)

- Une action à sous-activité « encapsule » un diagramme d'activité.
- Son usage est assez rare.



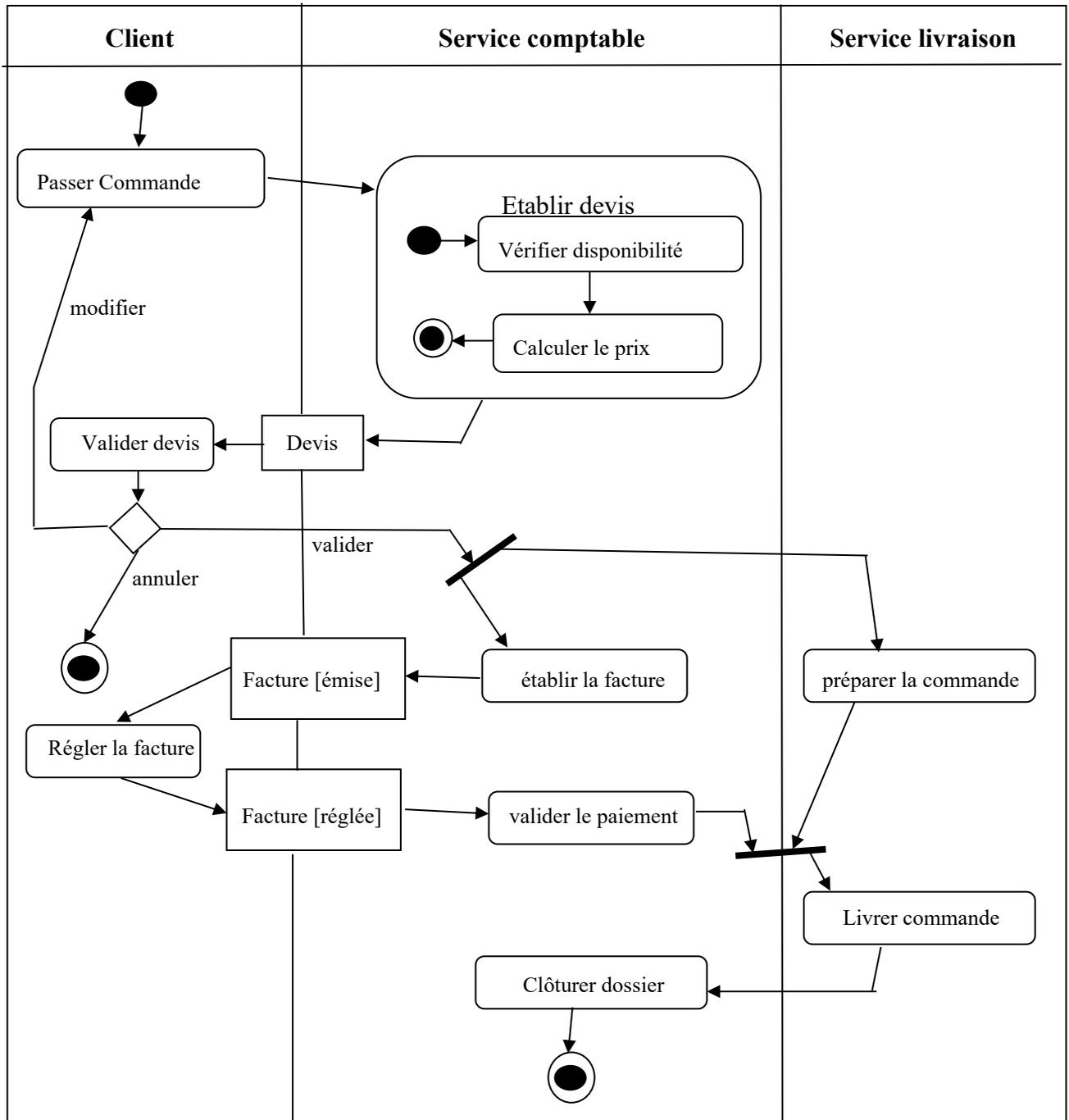
Élément particulier : diagramme d'activité à travées (slide 20)

- **Les travées** = couloir répartissent les actions par acteur réalisant les actions.
⇒ On représente un acteur par travée.
- Les acteurs :
⇒ des personnes ou des services.
⇒ Le plus souvent à un **poste de travail**
⇒ donc une **interface d'utilisation du système** (un téléphone portable, un site web, un client lourd, une machine dédiée, etc.).
- Les travées sont utilisées pour la description d'activité très générale (pas uniquement informatique).
- Dans les travées, on retrouve un diagramme d'activité classique.

Client	Service comptable	Service livraison

Rappel du slide 8

3 travées qui correspondent à 3 acteurs et potentiellement 3 postes de travail donc potentiellement 3 accès et usages possibles du système d'information.



● État initial

○ État final

Synchronisation ———

action

transition →

données

« Etablir devis » est une action à sous-activité.

3. Exercices

3-1 : Le save as de Word

Regardez le fonctionnement de l'UC « save as » de Word (ou de tout autre logiciel).

Faites le diagramme d'activité correspondant avec tous les cas possibles.

3-2 : Recette de cuisine : diagramme d'activité

Soit la recette suivante pour un gâteau au chocolat :

1. Préparer les ingrédients
2. Casser le chocolat en morceau et faites-le fondre
3. Pendant ce temps, séparer les blancs des jaunes.
4. Monter les blancs en neige.
5. Mélanger les jaunes, la farine et le sucre et le chocolat fondu.
6. Mélanger le résultat dans les blancs montés en neige.
7. Verser le mélange dans un plat
8. Mettez à cuire 40 minutes dans un four préchauffé à 140°.
9. Quand c'est cuit, démouler le gâteau.

Faite un **diagramme d'activité** décrivant la recette.

On intégrera le fait que si on n'a pas tous les ingrédients, on arrête la recette.

On précisera quand c'est utile les informations qui circulent dans les transitions.

3-3 : Arrivée d'un nouvel employé : diagramme d'activité à travées

Suite à un processus de recrutement d'un candidat dans une entreprise, le service des ressources humaines envoie une proposition d'embauche au candidat et attend sa réponse. Si la réponse du candidat est négative, le processus est stoppé. Sinon, le service des ressources humaines prépare les documents d'embauche et les font signer au candidat. En même temps, il prévient le service informatique d'ouvrir un compte et prévient le service logistique d'allouer un bureau et un téléphone.

1. Faire un **diagramme d'activité à travée** décrivant la situation.

3-4 : Diagramme d'activité des scénarios d'un UC

Principe

A partir d'un diagramme de cas d'utilisation (ceux des exercices (cf cours UC), du cours ou d'un projet), prendre les UC et, pour chaque UC, faire le **diagramme d'activité** des scénarios de l'UC.

La médiathèque

Sujet complet :

<http://bliaudet.free.fr/IMG/pdf/ESGI-UML2-4-1-Use-Case-23s.pdf#page=29>

- Diagramme d'activité de l'UC emprunter des livres
- Diagramme d'activité de l'UC rendre un livre
- Diagramme d'activité de l'UC réserver un livre

Les enchères

Sujet complet :

<http://bliaudet.free.fr/IMG/pdf/ESGI-UML2-4-1-Use-Case-23s.pdf - page=26>

A partir du diagramme d'UC :

1. Faire le diagramme d'activité de l'UC « Mettre aux enchères un objet ».
2. Faire le diagramme d'activité de l'UC « enchérir ».
3. Faire de diagramme d'activité de l'UC « suivre un objet »

Etc.

LE DIAGRAMME DE SEQUENCE « SYSTEME »

*Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.
La méthode est dénuée de sens tant qu'elle est déconnectée du rapport au savoir.*

1. Le diagramme de séquence système (11 slides, de 21 à 31)

Contexte (slide 21)

- Les diagrammes de séquences peuvent être utilisés en analyse fonctionnelle et en analyse technique. Toutefois, ils sont surtout utilisés en analyse fonctionnelle.
- Il y a 4 diagrammes UML qu'on utilise en analyse fonctionnelle :

ANALYSE FONCTIONNELLE	Diagramme UML
	Diagramme de cas d'utilisation - UC
	Activité
	Séquence
	États-transitions

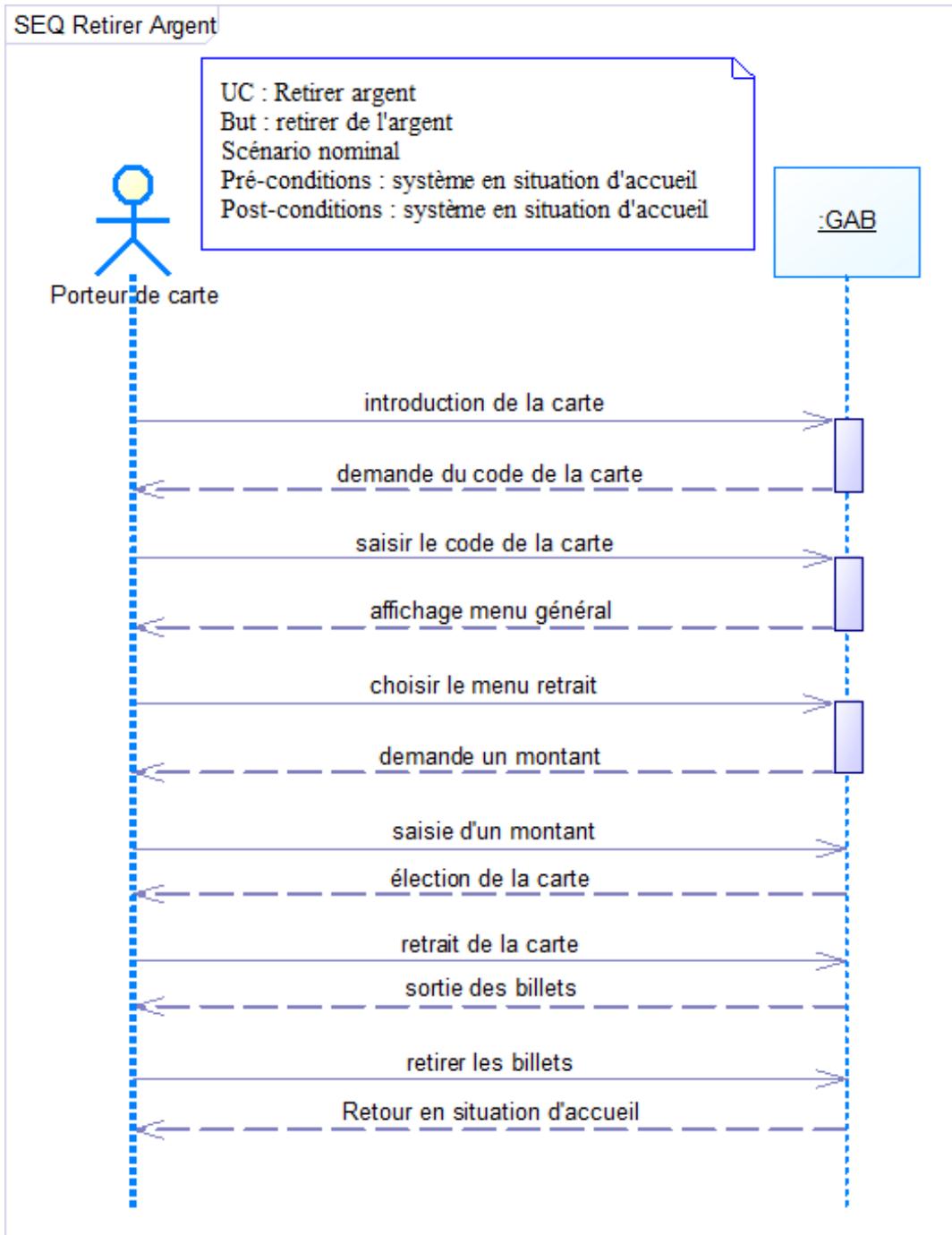
- On ne traite dans ce chapitre que du diagramme de séquence.

Principes

- Le diagramme de séquence « système » décrit les **échanges entre le système et les acteurs du système** (système = le logiciel à réaliser).
- Pour que l'analyse reste simple, **on analyse les échanges au niveau de chaque scénario nominal de chaque UC**.
 - ⇒ On peut donc avoir un diagramme de séquence système par UC : le DSS du SNUC
 - ⇒ Si on prend en compte tous les scénarios de l'UC, le diagramme de séquence système devient trop complexe.
- **C'est un diagramme simple** : il consiste à dérouler le scénario nominal de l'UC et à **se demander quelles informations circulent**, entre l'acteur et le système, séquentiellement, dans les 2 sens.
- **A ne pas confondre avec le diagramme de séquence « objet »** : le diagramme de séquence « objet » décrit les échanges de messages entre les objets, autrement dit, les appels de méthode d'un objet par un autre. Ce diagramme relève de l'analyse technique.
 - ⇒ Il sera abordé avec l'analyse MVC

Exemple : DSS du SNUC : retirer de l'argent au GAB (slide 23)

- On ne regarde que les interactions entre un acteur et le système.



Syntaxe du diagramme (slide 24 et 25)

➤ *Lecture du diagramme :*

- Le diagramme se lit de haut en bas en suivant les flèches.

➤ *L'acteur « porteur de carte » et sa ligne de vie*

- C'est celui qui démarre l'action. La première flèche part de lui.
- Le trait vertical pointillé sous l'acteur est une « **ligne de vie** » : on y décrit la succession des événements qui partent de l'acteur et arrivent à l'acteur. **Le début est en haut, la fin en bas.**

➤ *L'objet « GAB » : le système.*

- Il représente tout le système avec lequel l'acteur interagit.
- Le trait vertical pointillé sous le système GAB est une « **ligne de vie** » : on y décrit la succession des événements qui arrivent au GAB et qui partent du GAB. **Le début est en haut, la fin en bas.**

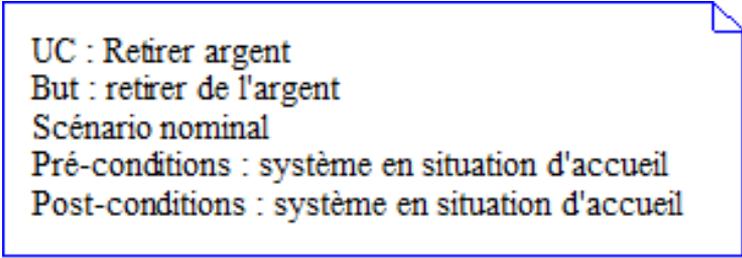
⇒ Souvent on écrit « Système » à la place de GAB ou autre chose.

⇒ En UML, « :GAB » vaut dire : un objet de la classe GAB. Les objets sont soulignés. Les noms de classe commencent par des majuscules. Ici, c'est un objet anonyme : il n'a pas de nom, et virtuel : la classe GAB n'existe pas. L'objet en question, c'est tout le système (tout le logiciel).

➤ *Les flèches, les boîtes de déroulement de code (slide 25)*

- La flèche en trait plein représente les demandes faites par l'acteur et les informations qu'il envoie au système.
- La flèche en pointillé représente le retour du système pour l'acteur suite à sa demande.
- Entre les deux, le rectangle côté système matérialise le déroulement d'une action du système, depuis la demande par l'acteur jusqu'au retour par le système. On peut ne pas mettre ce rectangle dans le schéma.

➤ *Le descriptif de l'UC*

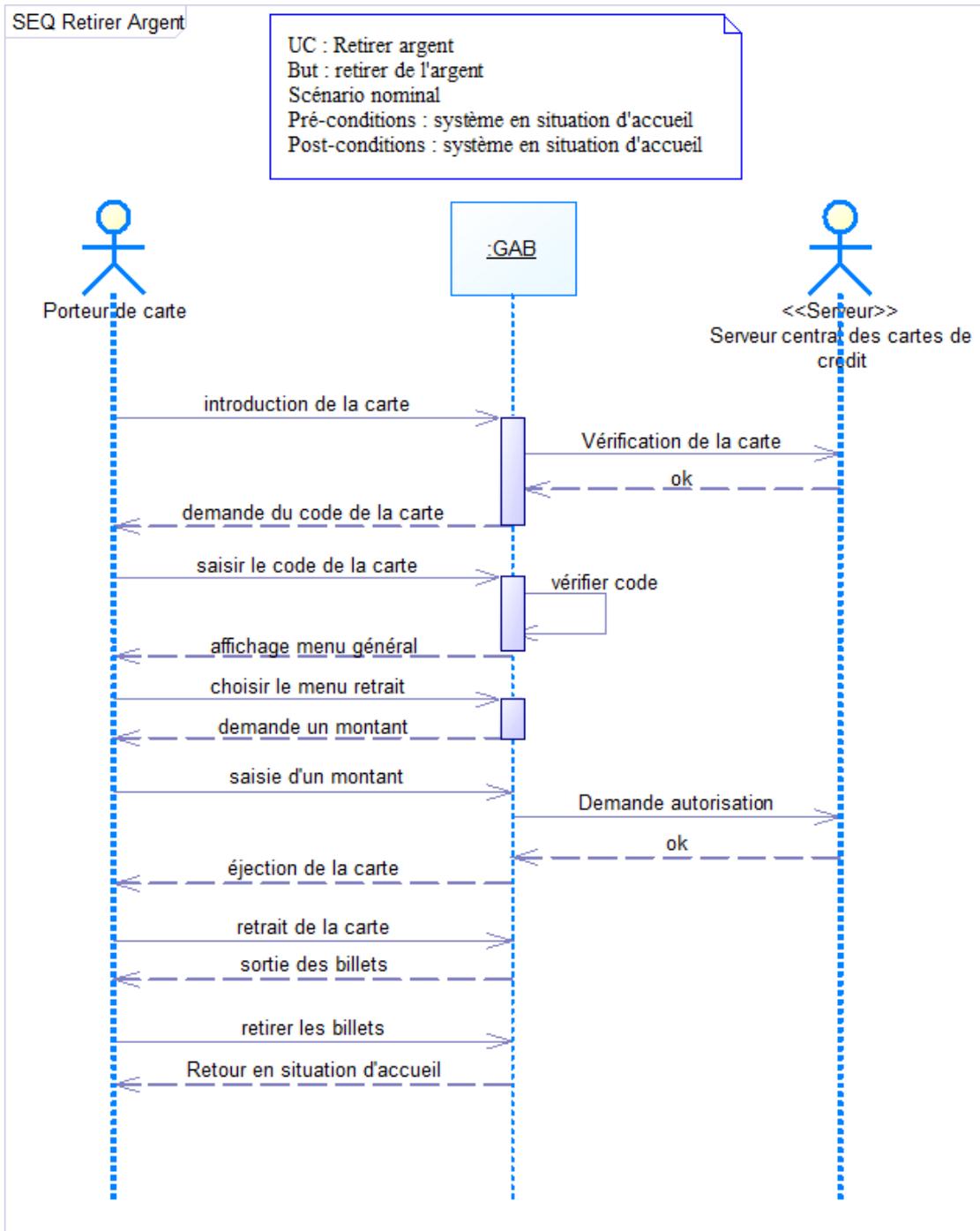


UC : Retirer argent
But : retirer de l'argent
Scénario nominal
Pré-conditions : système en situation d'accueil
Post-conditions : système en situation d'accueil

- On précise l'UC, le but de l'UC et le ou les scénarios concernés. En général, le scénario nominal.
- Les **pré-conditions** précisent l'état du système avant la mise en œuvre de l'UC.
- Les **post-conditions** précisent l'état du système après le déroulement de l'UC.
⇒ Dans ce cas : SNUC, pas de pré et post conditions : le descriptif est inutile.

Diagramme de séquence système avec des acteurs passifs (slide 26)

- On peut ajouter les acteurs passifs dans le diagramme de séquence système.
- Le diagramme est plus complexe mais aussi plus explicite.

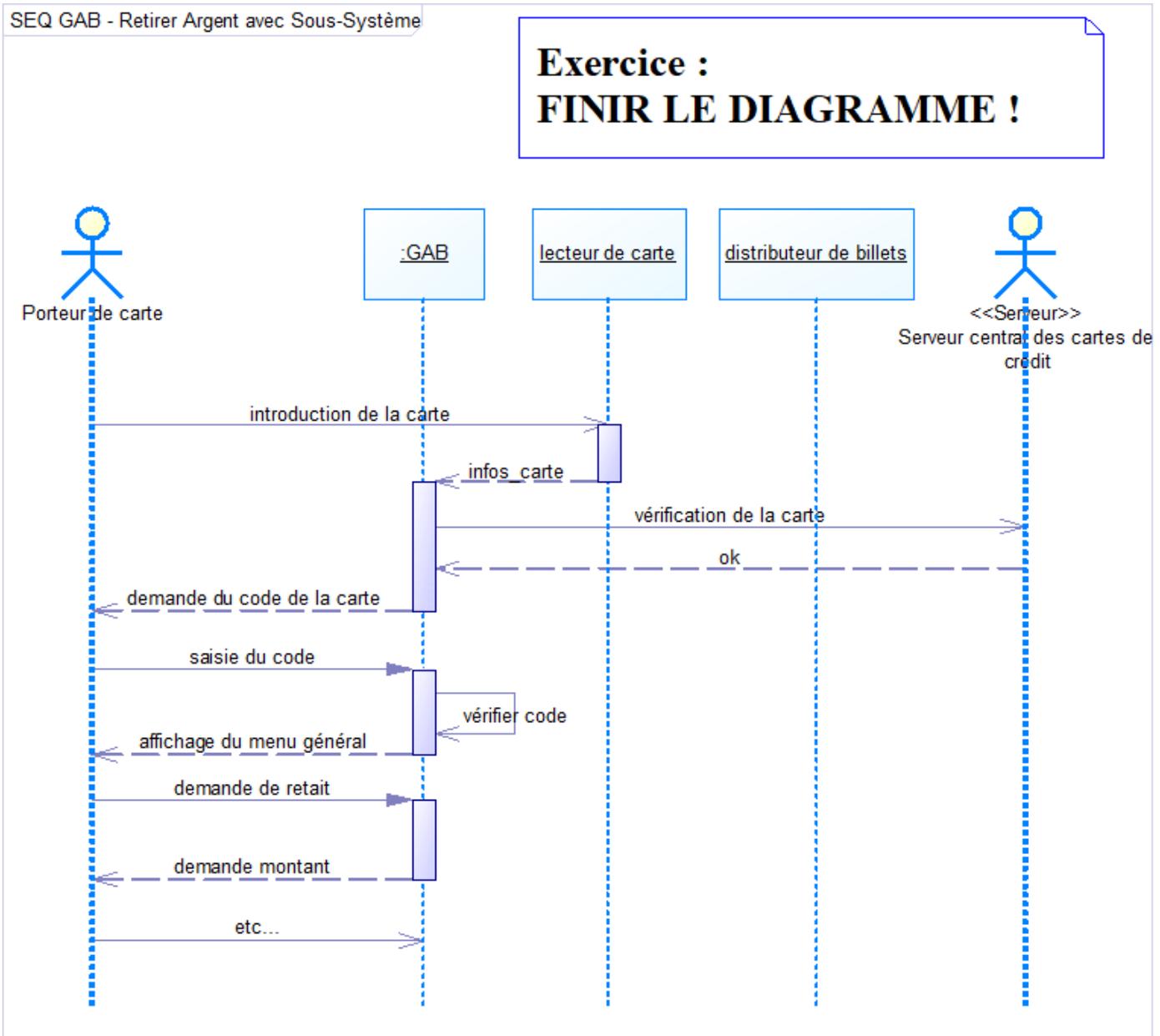


Syntaxe du diagramme (slide 27)

- *Les interactions avec un acteur passif à partir d'une action du GAB*
 - Une flèche en trait plein de demande de vérification ou d'autorisation. Une flèche en pointillée de réponse de l'acteur passif avec l'information retournée, ici « ok ».
 - Les flèches partent et arrivent du rectangle de déroulement d'une action du GAB, même si le rectangle n'est pas représenté.
- *Une flèche réflexive pour signaler une action en interne*
 - La vérification du code est faite en interne. On la précise pour montrer qu'elle ne fait pas appel à un acteur externe.

Diagramme de séquence système avec des sous-systèmes (slide 28)

- On peut ajouter les sous-systèmes dans le diagramme de séquence système.
- Ça nécessite d'avoir analysé les sous-systèmes.
- Le diagramme est plus complexe mais aussi plus explicite. Il clarifie particulièrement le fonctionnement des sous-systèmes.



Syntaxe du diagramme (slide 29)

➤ *Les sous-systèmes*

- Le distributeur est composé de plusieurs parties : l'interface utilisateur, le lecteur de carte, la partie qui sort les billets (au moins).
- Les sous-systèmes interagissent entre eux pour produire le résultat final : un UC.
- Les sous-systèmes sont représentés comme le système : ce sont des objets. On met directement le nom du sous-système, en minuscule et souligné, sans préciser de classe d'appartenance.
- A partir de là, on retrouve les interactions classiques d'un diagramme de séquence système.
- Les flèches partent et arrivent du rectangle de déroulement d'une action du GAB, même si le rectangle n'est pas représenté.

Variante à éviter : Description textuelle des UC

Principe

- La description textuelle permet de représenter les alternatives et les répétitions.

Défaut

- C'est lourd et pas pratique.

Description textuelle du UC : retirer de l'argent.

Identification	
Nom du cas :	retirer de l'argent
But :	retirer de l'argent dans un GAB
Acteur principal :	le porteur de carte
Acteurs secondaires :	le système central
Pré-conditions (état du système avant l'opération) :	Ecran d'accueil affiché
Post-conditions (état du système après l'opération) :	Ecran d'accueil affiché
Enchaînement nominal	
1.0	<u>Le porteur de carte introduit sa carte</u>
2.0	<i>Le système demande le code de la carte</i>
3.0	<u>Le porteur de carte saisit le code de la carte</u>
4.0	<i>Le système affiche le menu général</i>
5.0	<u>Le porteur de carte choisit le retrait</u>
6.0	<i>Le système demande le montant du retrait</i>
7.0	<u>Le porteur de carte choisit un montant</u>
8.0	<i>Le système éjecte la carte</i>
9.0	<u>Le porteur de carte retire la carte</u>
10.0	<i>Le système éjecte les billets</i>
11.0	<u>Le porteur de carte retire les billets</u>
FIN	<i>Le système affiche l'écran d'accueil</i>
Alternatives n°2.0	
2.1.1	Le système rejette la carte qui n'est pas valide
2.1.2	Goto FIN : le système affiche l'écran d'accueil

2.2.1	Le système conserve la carte qui est interdite et affiche un message
2.2.2	Affichage de l'écran d'accueil
Alternative n°4.0	
4.1.1	Le code est erroné et le système affiche un message
4.1.2	Goto 2.0 : le système demande le code de la carte

4.2.1	Le code est erroné 3 fois de suite, le système garde la carte et affiche un message
4.2.2	Goto FIN : le système affiche l'écran d'accueil

4.3.1	Le porteur de carte a annulé la saisie.
4.3.2	<i>Le système éjecte la carte</i>
4.3.3	<u>Le porteur de carte retire la carte</u>
4.3.4	<i>Le système affiche l'écran d'accueil</i>
Alternative n°8.0	
8.1.1	Le montant est trop élevé. Le système affiche un message.
4.1.2	Goto 6.0 : le système demande le montant du retrait
Etc.	

2. Variantes à éviter : Modularité, test, boucle, autres contraintes

Modularité : à éviter

Principes

- On peut mettre des interactions répétitives dans un bloc, comme dans une fonction.
- On peut faire appel à ce bloc dans nos futurs diagrammes.

Intérêt

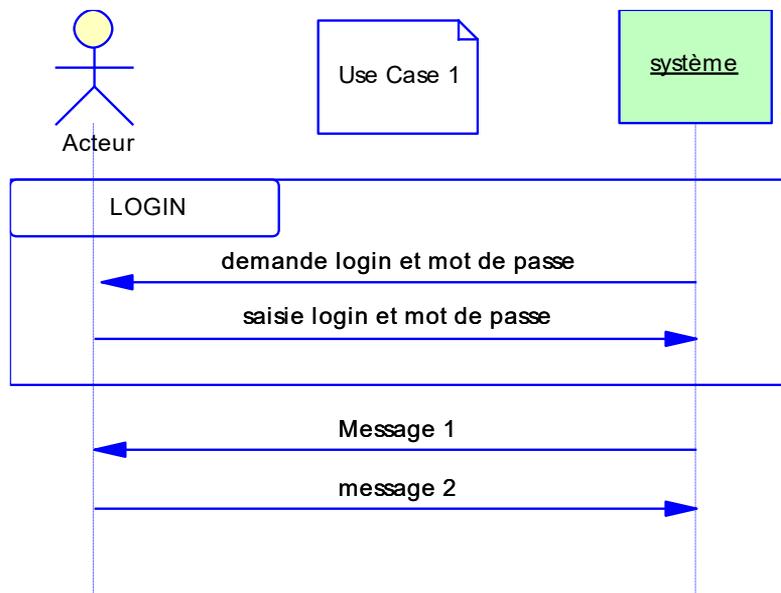
- Comme pour toute fonction, ça factorise l'analyse.

Défaut

- C'est très technique. Trop technique, donc inutile car la réalisation peut changer beaucoup d'élément d'interface.
 - ⇒ A ce stade, ce qui nous intéresse c'est de faire le « fonctionnel » du système. L'analyse des SNUC suffira.

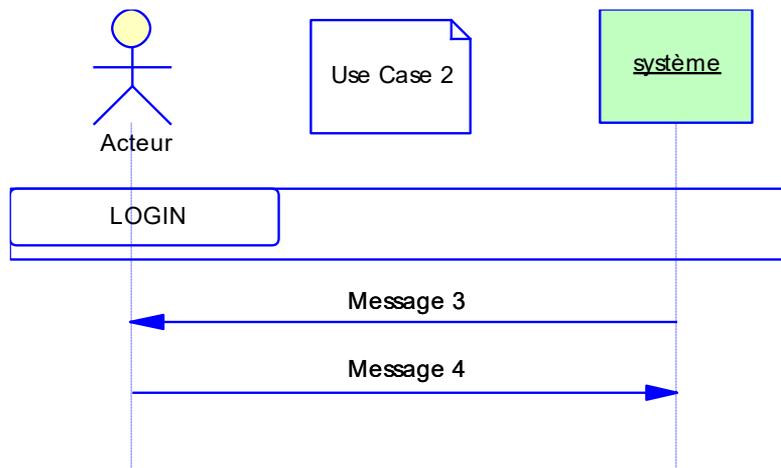
Exemple et syntaxe :

➤ *Diagramme de séquence du Cas d'utilisation 1 :*



➤ *Diagramme de séquence du Cas d'utilisation 2 :*

On réutilise le module « LOGIN » :



Présentation

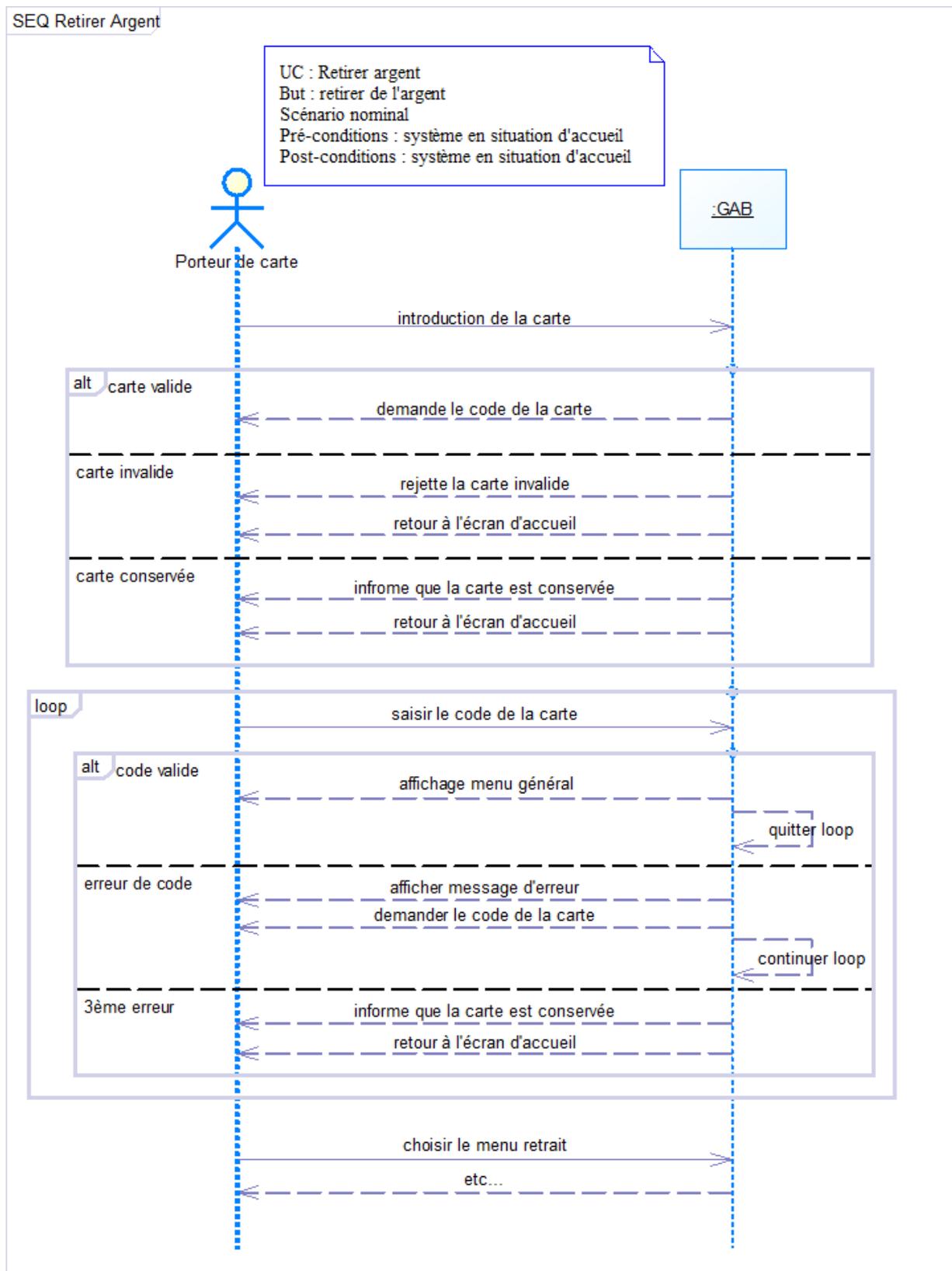
On peut représenter des **tests** et des **boucles** dans les diagrammes de séquence.

Le diagramme devient très complexe : c'est à éviter !!!

<http://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-interaction>

Exemple : scénarios alternatifs d'un UC dans le DSS

- On peut traiter tous les scénarios d'un UC dans le diagramme de séquence système.



Syntaxe du diagramme

➤ *La boîte « alt » (alternative)*

- Elle permet de définir des séquences alternatives. On précise la condition et on met les échanges pour chaque condition. Dans l'exemple : si la case est minée, si la case est numérotée, dans les autres cas.
- On peut préciser la condition à côté du « alt ».

➤ *La boîte « opt »*

- Elle permet de n'avoir qu'une alternative (un si sans sinon : c'est l'équivalent d'un extend dans les UC)

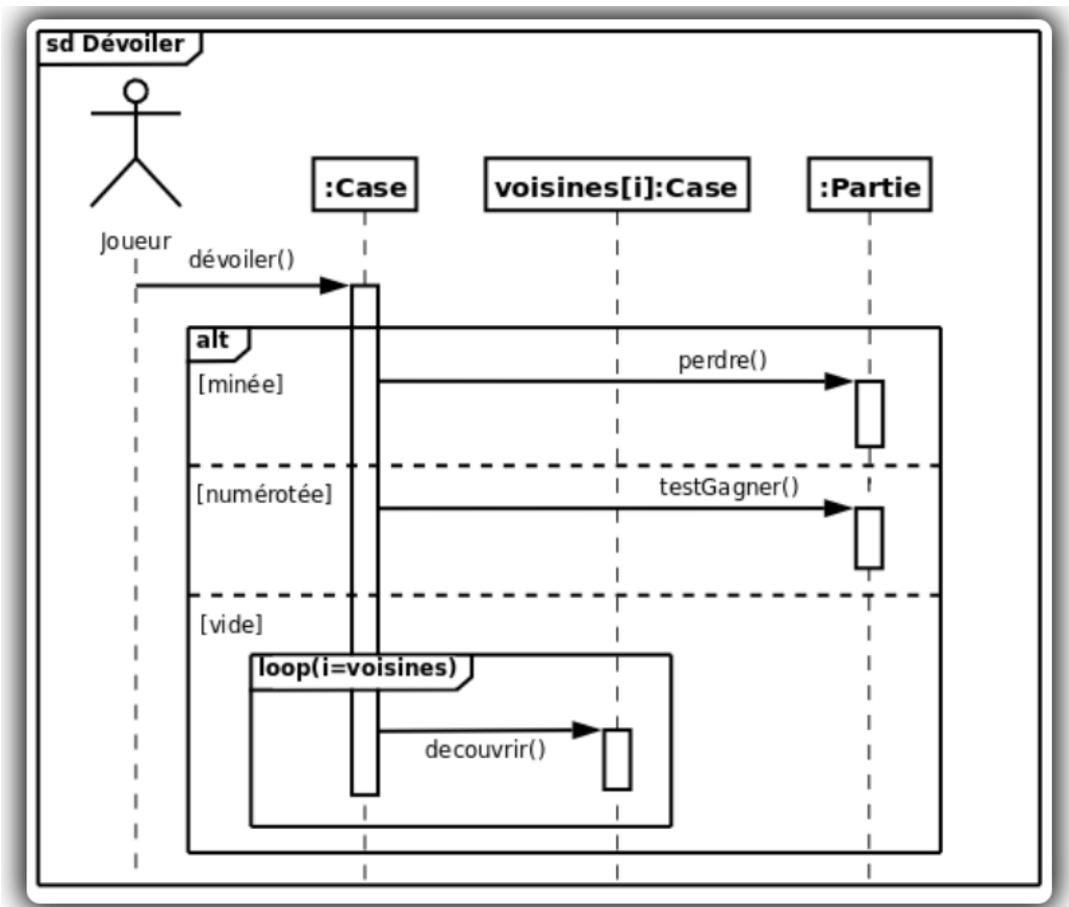
➤ *La boîte « loop »*

- Une boîte « loop » permet de représenter une répétition (une boucle).
- On peut préciser toute caractéristique de boucle :
 - ⇒ un compteur
 - ⇒ des messages « quitter boucle » et « continuer boucle »

➤ *Autres possibilités*

- Les logiciels proposent plus ou moins d'autres possibilités.

Exemple avec un diagramme de séquence objet



Autres contraintes, commentaires et notes

On peut représenter toutes sortes de contraintes par des **commentaires** : des **notes attachées aux éléments graphiques qu'on veut contraindre**.

3. Utilisation en architecture Système et Réseaux

Communication entre les serveurs (slide 30)

- On peut utiliser le DSS pour décrire un environnement « système et réseaux ».
 - ⇒ Pour montrer la communication entre les différents serveurs :
 - OS, serveur d'authentification, serveur d'émission de tickets, etc.

Cet usage est malheureusement peu répandu chez les ingénieurs système et réseaux !

Au lieu de cela, il utilise des schémas non standard et qui en général mélangent plusieurs niveaux d'information, rendant les choses peu claires !

Exemple : protocole d'authentification Kerberos (slide 31)

[https://fr.wikipedia.org/wiki/Kerberos_\(protocole\)](https://fr.wikipedia.org/wiki/Kerberos_(protocole))

Solution non UML

Ce schéma qui ressemble à un diagramme de séquence système mais mélange plusieurs niveaux d'information :

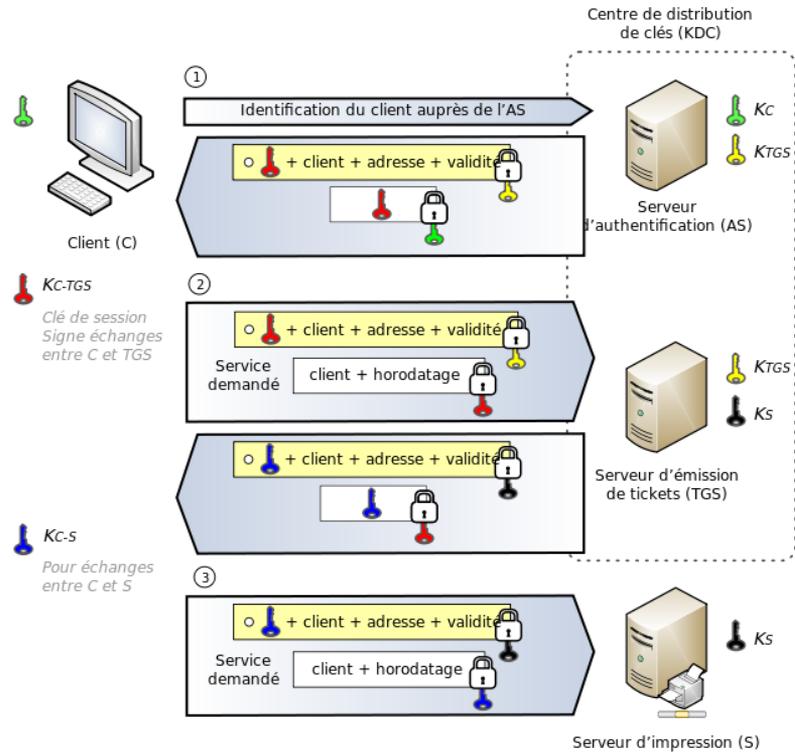


Diagramme UML : diagramme de séquence système

Mieux vaudrait un diagramme de séquence système ! Mesdames et messieurs les futurs ingénieurs système : à vous de jouer !

4. Exercices

4-1 : DSS du SNUC

Principe

A partir d'un diagramme de cas d'utilisation (ceux des exercices (cf cours UC), du cours ou d'un projet), prendre les UC et, pour chaque UC, faire le **Diagramme de Séquence Système** du **Scénario Nominal** de l'UC : **DSS du SNUC**

La médiathèque

Sujet complet :

- <http://bliaudet.free.fr/IMG/pdf/ESGI-UML2-4-1-Use-Case-23s.pdf#page=29>
- A partir du diagramme d'UC :
 - DSS du SNUC emprunter des livres
 - DSS du SNUC rendre un livre
 - DSS du SNUC réserver un livre

Les enchères

Sujet complet :

<http://bliaudet.free.fr/IMG/pdf/ESGI-UML2-4-1-Use-Case-23s.pdf - page=26>

- A partir du diagramme d'UC :
 - DSS du SNUC « Mettre aux enchères un objet ».
 - DSS du SNUC « enchérir»
 - DSS du SNUC « suivre un objet»

etc.

4-2 : Protocole d'identification Kerberos

- Le protocole est décrit ci-dessus au chapitre 3.
 - ⇒ Traduisez-le par un diagramme de séquence.

DIAGRAMME D'ETATS-TRANSITIONS

*Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.
La méthode est dénuée de sens tant qu'elle est déconnectée du rapport au savoir.*

1. Présentation (11 slides, de 32 à 42)

Contexte (slide 32)

- Les diagrammes d'états-transitions peuvent être utilisés en analyse fonctionnelle et en analyse technique.
- Leur usage est le même qu'on soit au niveau fonctionnel ou technique
- Il y a 4 diagrammes UML qu'on utilise en analyse fonctionnelle et 6 diagrammes qu'on utilise en analyse organique.

	Point de vue	Diagramme UML
ANALYSE FONCTIONNELLE	Statique – non objet Analyse générale	Cas d'utilisation
	Dynamique – non objet Analyse détaillée	Séquence
		Activité
		États-transitions
ANALYSE ORGANIQUE	Statique – objet	Classes
		Objets
	Dynamique – objet	Séquence
		Communication
	Dynamique Plutôt non objet	États-transitions
	Activité	

- On ne traite dans ce chapitre que du diagramme d'états-transitions

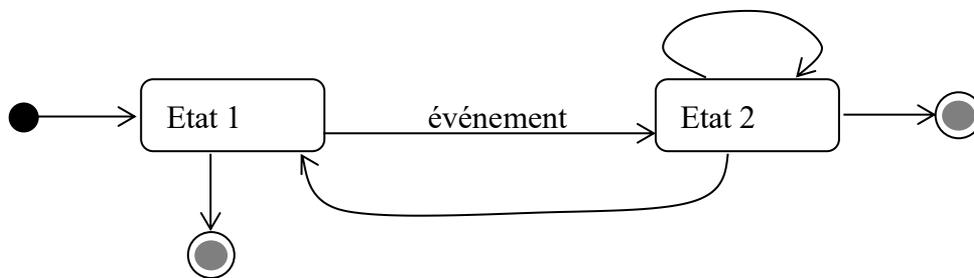
Source Web :

<https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-etats-transitions>

Principes (slide 33)

- Le digramme d'états-transitions montre les différentes valeurs d'une information pendant le déroulement du programme.
- La valeur de cette information c'est son « état ».
- L'information peut être rangée dans une ou plusieurs variables ou dans un ou plusieurs attributs d'un objet.
- Le passage d'une valeur à une autre (d'un état à un autre) se fait via une transition.
- La transition correspond à une action qui change l'état.
- L'action correspond à un événement déclencheur qui change l'état.
- A la place de diagramme d'états-transitions, on parle aussi de diagramme d'états ou de graphe d'états.

Représentation UML

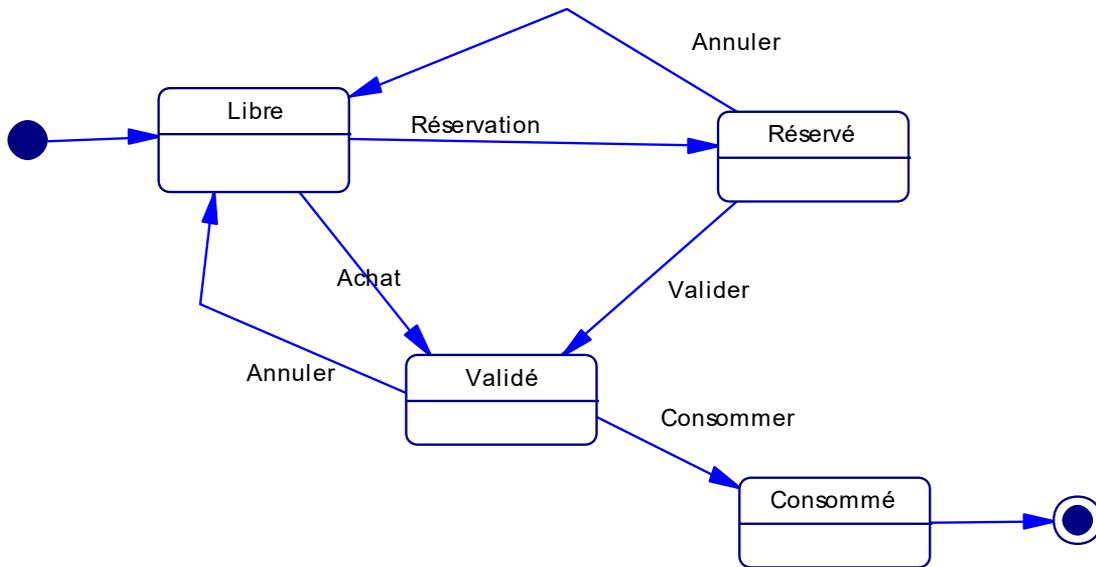


- L'état est un rectangle avec des angles arrondis.
- L'état initial est un **point noir**. Il n'y a en général qu'un seul, mais il peut y en avoir plusieurs.
- L'état final est un **point noir entouré d'un cercle**.
- Les flèches sont des transitions. L'événement est le déclencheur de la transition. Il peut être conditionné.

Réservation

- On modélise l'état de place de spectacle. L'état d'une place de spectacle peut être : libre, réservé, validé, consommé.
- Plusieurs action-événements sont pris en compte : l'achat, la réservation, la validation, l'annulation, la « consommation » de la place (son utilisation effective).
- La réservation n'est pas un achat. Elle peut être annulée.
- On peut acheter directement une place sans l'avoir à la réserver. Les places achetées sont dites validées. On pourra aussi la valider après la réservation. On pourra aussi annuler une place validée.
- Seules les places validées (et donc achetées) peuvent être consommées.
- On peut présenter un ou plusieurs points de départ et un ou plusieurs points finaux.

Solution (slide 35)



Bilan

- L'intérêt du diagramme est de rendre les choses plus claires qu'avec un texte.

2. Syntaxe des diagrammes d'états-transitions - vocabulaire et graphique

État (slide 36)

État (ou état effectif)

- Un état effectif (ou état) est une **situation donnée durant la vie d'un objet**. C'est une situation plus ou moins durable.
- Cela correspond à un **ensemble de valeurs pour un ensemble d'attributs** (une seule valeur pour un seul attribut dans le cas le plus simple).
- Un état peut être :
 1. Initial
 2. Intermédiaire
 3. Final
- Les états intermédiaires et l'état final dépendent des états précédents et des événements survenus.

État initial (slide 37)

- L'état initial est un **pseudo-état** relié au premier état effectif par une **pseudo-transition**.
- Dans un diagramme d'états-transitions, il peut y avoir **plusieurs états initiaux**.

Premier état

- Un premier état est un état effectif qui **suit un état initial**. Il peut y en avoir plusieurs.

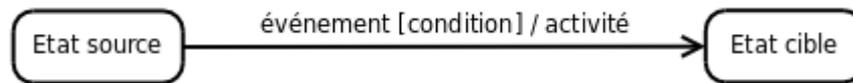
État final

- C'est un **pseudo-état** relié au dernier état effectif par une **pseudo-transition**.
- Dans un diagramme d'états-transitions, il peut y avoir **0 ou plusieurs états finaux** (0 dans le cas d'un système qui ne s'arrête jamais).

Dernier état

- Un dernier état est un état effectif qui **précède un état final**. Il peut y en avoir plusieurs, ou aucun.

Transition



- Une transition relie deux états de manière unidirectionnelle : un et un seul état initial et un et un seul état final.
- Un état peut être la source de plusieurs transitions et la cible de plusieurs transitions.
- Les transitions indiquent les chemins possibles dans les graphes des états.
- Une transition correspond à des actions à réaliser entre deux états.

Événement

- Un événement déclenche une transition qui fait passer d'un état à un autre.
- Un événement est une **situation qui ne dure pas, contrairement à un état**. La validation d'un achat sur internet, c'est un événement. L'enregistrement d'un état « validé » en BD, c'est un état qui dure ;
- L'événement est **forcément précisé sur le diagramme** (pas la condition ni l'activité).

Condition (ou condition de garde)

- La condition **conditionne le déclenchement de la transition** lors de l'occurrence d'un événement. Il peut ne pas y en avoir.

Activité ou effet

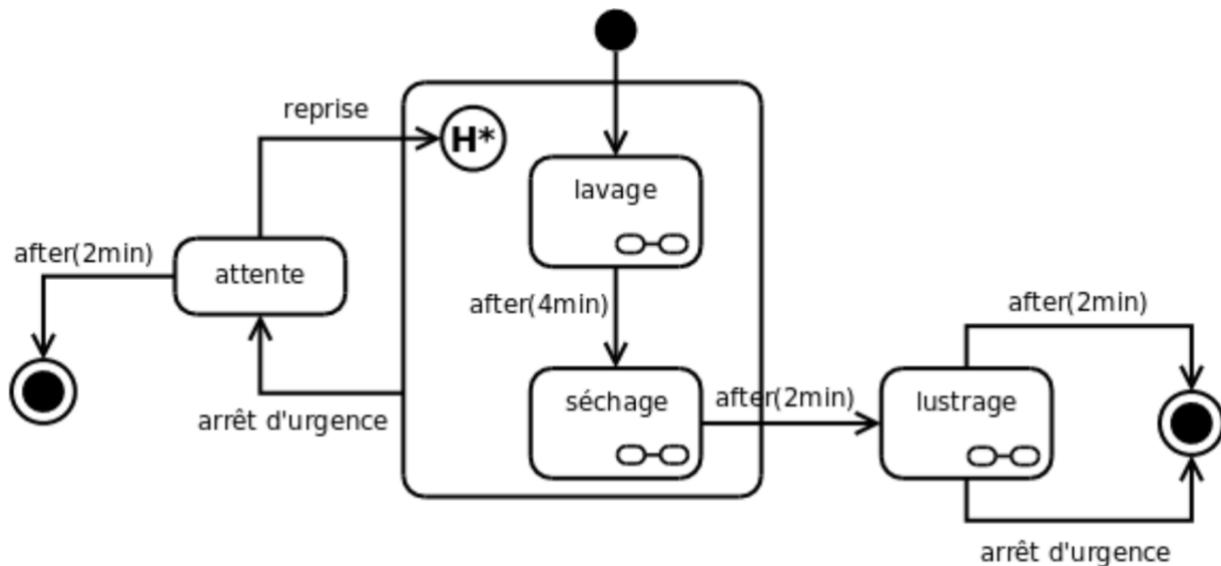
- Une transition a généralement un effet : c'est une activité qui s'exécute après que l'événement déclencheur ait lieu si la condition soit réalisée.

3. État composite (slide 39)

Principe

- Un état composite est donc un état qui contient un diagramme d'états-transitions interne en son sein.
- **Exemple d'usage : un état e1 peut correspondre à un objet qui lui-même peut avoir différents états.** En général, cet objet est un composant de l'objet de départ. Dans l'état e1 est dit composite.
- **Autre exemple d'usage : on peut vouloir regrouper des états dans un état composite** pour appliquer à l'état composite des propriétés communes aux états regroupés.

Exemple de regroupement :



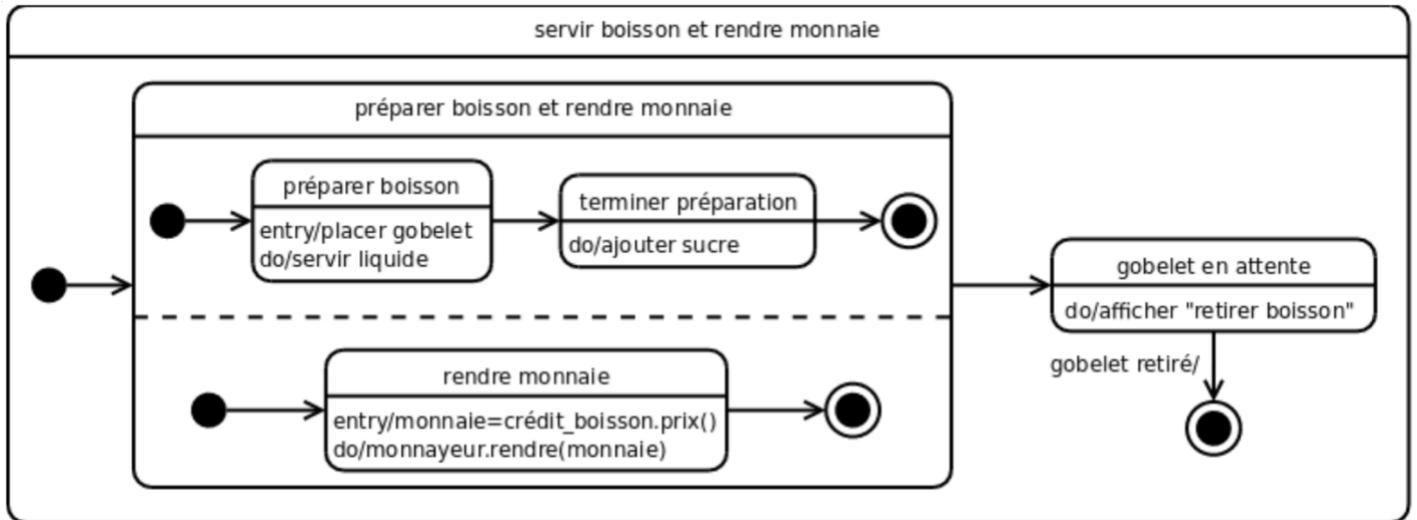
Historique

- Le H* veut dire qu'on peut quitter l'état composite dans n'importe quel état (lavage ou séchage) et qu'on y reviendra dans l'état où on l'a laissé.

État concurrent (slide 40)

- Un système peut gérer plusieurs diagrammes d'états en parallèle (on dit aussi en concurrence).
- On peut les présenter l'un en dessous de l'autre en les séparant par un trait en pointillé.
- On peut les mettre dans un rectangle d'état en les séparant par un trait en pointillé.

Exemple



Remarque

A noter que les états correspondent ici directement à des actions. Le diagramme d'état est très proche d'un diagramme d'activité.

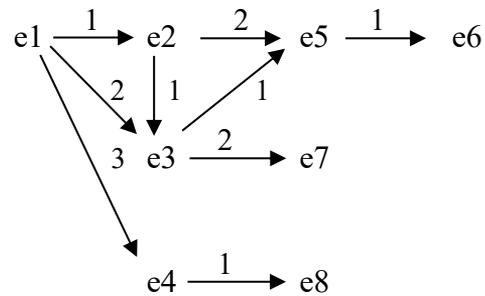
4. Automate d'états finis (slide 41)

Principe

- Un graphe d'états-transitions peut être transformé en un « automate d'états finis », aussi appelé aussi simplement « automate » ou « automates d'états » ou bien « machine à états ».
- Cet automate se traduit par un tableau à 3 colonnes : état initial, état final, transition.

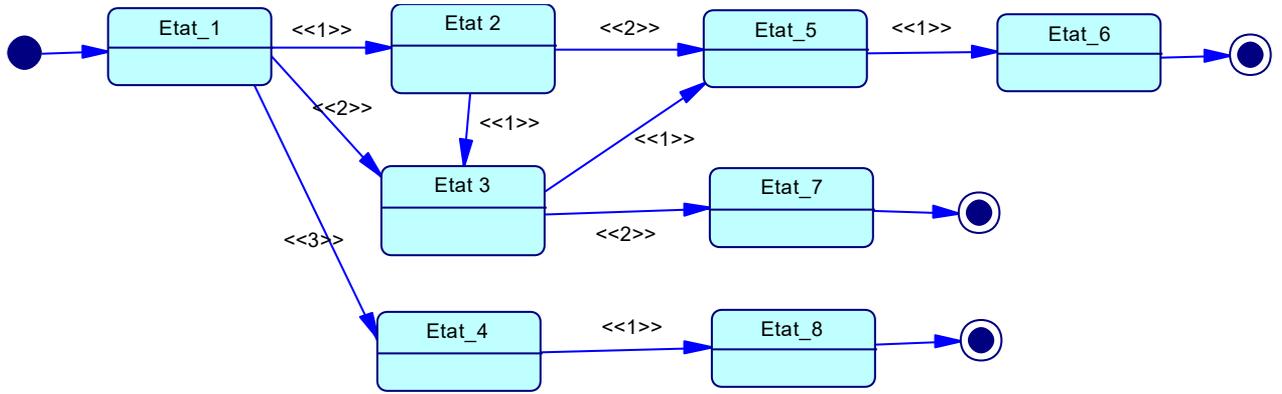
Exemple

➤ Graphe d'états



- Ce graphe nous montre que pour passer :
 - ⇒ de e1 à e2, il faut l'événement 1,
 - ⇒ de e2 à e3, il faut l'événement 1 aussi,
 - ⇒ de e1 à e2, il faut l'événement 2,
 - ⇒ etc.

➤ **Grphe des états, version UML (slide 42)**



➤ **Automate d'états finis**

Etat initial	Etat final	Transition
1	2	1
1	3	2
1	4	3
2	3	1
2	5	2
3	5	1
3	6	2
4	7	1
5	8	1

5. Exercices

5-1 : La porte de garage

Soit une porte de garage à enroulement motorisé. On peut manœuvrer cette porte avec une télécommande avec un seul bouton. Une pression sur le bouton à les effets suivants sur la porte :

- L'ouvrir si elle est fermée
- La fermer si elle est ouverte
- L'ouvrir si elle est en cours de fermeture
- La fermer si elle est en cours d'ouverture.

Un capteur de butée sur la porte indique si la porte est ouverte (en position haute) ou fermée (en position basse).

4. Quels sont les objets et leurs états possibles ?
5. Quels sont les événements qui peuvent advenir ?
6. Faire un diagramme d'états-transition
7. Déterminer le ou les états initiaux et finaux possibles.
8. Comment faire en sorte que si on remet la porte sous tension alors qu'elle qu'on l'a éteinte en cours de fermeture qu'elle ne se referme pas si on clique sur la télécommande ?
9. Une fois la solution à la question précédente trouvée, dite ce qu'il se passe, en fonction de l'état de la porte

5-2 : La montre digitale

Soit une montre digitale avec 4 boutons. Modéliser successivement un diagramme d'états.

Le mode courant de la montre est l'affichage de l'heure. La montre a 3 modes : « affichage », « réglage d'heure », « chronomètre ». En appuyant sur le bouton « mode » (bouton 1), on passe d'un mode à l'autre puis on revient à l'affichage de l'heure. L'utilisateur dispose de 2 minutes pour régler l'heure. Passé ce délai, la montre revient en mode « affichage »

L'affichage de l'heure est actualisé toutes les secondes.

Le bouton light (bouton 2) permet d'éclairer montre quel que soit le mode. Au bout de 5 secondes, l'éclairage s'arrête. Une pression sur le bouton light en cours d'éclairage réinitialise la durée de l'allumage.

En mode chronomètre, le bouton « start/stop » (bouton 3) lance, arrête ou relance le chronomètre. Le bouton « set » (bouton 4) remet le chronomètre à 0 s'il est arrêté.

L'affichage du chronomètre est actualisé tous les centièmes de secondes.

Quand le chronomètre est lancé, on peut changer de mode et que le chronomètre tourne toujours.

En mode « réglage », le bouton « start/stop » permet de passer du réglage des heures à celui des minutes, puis des secondes puis à nouveau des heures. L'afficheur fait clignoter l'élément concerné. Le bouton « set » permet d'incrémenter les heures et les minutes, et de remettre les secondes à zéro. L'heure affichée continue de courir pendant le réglage.

L'appui sur le bouton « set » pendant 2 secondes permet de faire défiler les heures ou les minutes rapidement jusqu'à relâche du bouton