

La modélisation dynamique

« haut niveau »

ACDA – CPOO (M3105)

Mathieu Sassolas

IUT de Sénart Fontainebleau
Département Informatique

Année 2015-2016
Cours 2



Plan de la séance

Scenarii et séquence

M. Sassolas
M3105
Cours 2

Programme

Scenarii

Séquence

Activités

2 / 47

- 1 Programme This is so meta !
- 2 Scénarios textuels
 - Pourquoi ?
 - Comment ?
- 3 Diagrammes de séquence
 - Syntaxe (révisions IS2)
 - Diagrammes de séquence « système » ?
- 4 Diagramme d'activités (si on a le temps)
 - Syntaxe (révisions IS2)
 - Compliquons : ajoutons des objets

Plan de la séance — En image

Scenarii et séquence

M. Sassolas
M3105
Cours 2

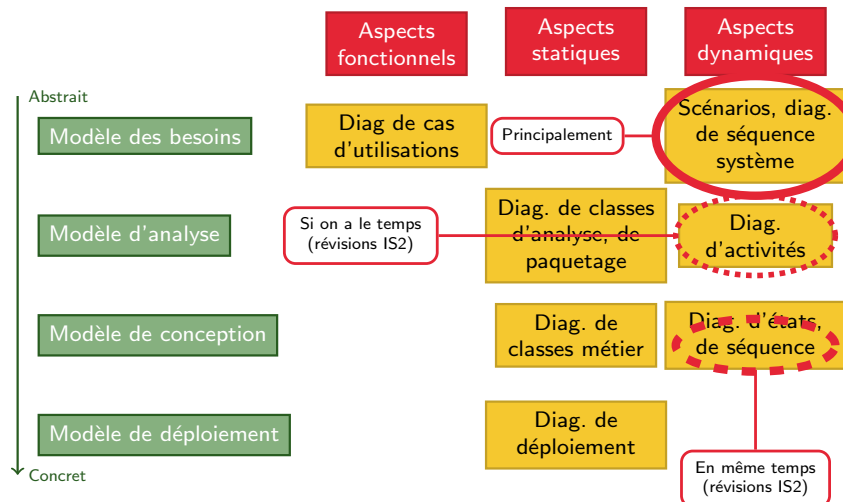
Programme

Scenarii

Séquence

Activités

3 / 47



Plan de la séance

Scenarii et séquence

M. Sassolas
M3105
Cours 2

Programme

Scenarii

Pourquoi ?
Comment ?

Séquence

Activités

4 / 47

- 1 Programme This is so meta !
- 2 Scénarios textuels
 - Pourquoi ?
 - Comment ?
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)

- 1 Programme This is so meta !
- 2 Scénarios textuels
 - Pourquoi ?
 - Comment ?
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)

- ▶ Identifier les **interactions** du système avec son environnement : **humains** et **autres systèmes**.
- ▶ Identifier les **besoins** : fonctionnalités du système.
- ▶ Identifier les **dépendences** entre les fonctionnalités.



Ça ressemble aux cas d'utilisation !



Cours 1,
diapo 13

Rappel (Cours 1, diapo 31)

Un diagramme de cas d'utilisation ne donne pas d'information sur le temps.

Grâce aux scénarios

On va pouvoir spécifier l'enchaînement temporel des cas et des interactions des acteurs avec le système.

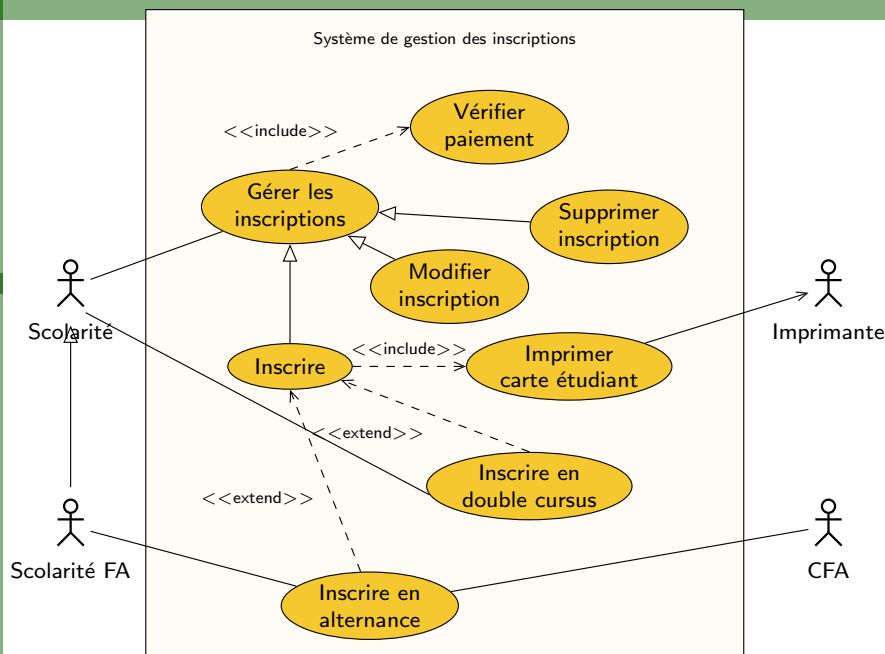
Et en plus

Ils vont permettre de préciser ce qui est entendu dans le nom des cas d'utilisation.

- 1 Programme This is so meta !
- 2 Scénarios textuels
 - Pourquoi ?
 - Comment ?
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)

- Titre** Le cas d'utilisation qu'on est en train de décrire.
- Objectif** Une description succincte du **but** du cas d'utilisation (le **pourquoi**).
- Acteurs** Les **acteurs** impliqués ; si l'un d'eux démarre l'interaction, on le précise en tant qu'**initiateur**.
- Type** On distingue le **scénario nominal** (ce qui se passe « normalement ») des **scénarios d'erreur**.
- Préconditions** Ce que l'on suppose vrai pour l'exécution du scénario.
- Postconditions** Ce qui sera vrai après l'exécution du scénario.

- Titre** Inscrire en alternance
- Objectif** L'étudiant concerné sera inscrit à l'université en tant qu'apprentis.
- Acteurs** La scolarité FA (initiateur), le CFA.
- Type** Scénario nominal.
- Préconditions** L'étudiant a déjà une entreprise, l'étudiant a payé ses frais d'inscription.
- Postconditions** L'étudiant est inscrit et sa carte d'étudiant est imprimée.



- Titre** Inscrire en alternance
- Objectif** L'étudiant concerné sera inscrit à l'université en tant qu'apprentis.
- Acteurs** La scolarité FA (initiateur), le CFA.
- Type** Scénario nominal.
- Préconditions** L'étudiant a déjà une entreprise, l'étudiant a payé ses frais d'inscription.
- Postconditions** L'étudiant est inscrit et sa carte d'étudiant est imprimée.

Descriptif Donne en français la suite des étapes du scénario sous la forme d'une liste numérotée.

Flux alternatifs Indiquent les endroits où des erreurs pourraient se produire.

Cas référencés Lorsqu'un cas d'utilisation dépend (**extend** ou **include**) d'un autre, on peut ne pas détailler le sous-cas dans le descriptif, mais on mentionne que l'on utilise ce sous cas.

Descriptif

1. La scolarité FA crée un dossier concernant l'étudiant.
2. La scolarité FA remplit les informations de l'étudiant.
3. La scolarité FA valide le dossier.
4. Le système transmet le dossier au CFA.
5. Le CFA établit un contrat de travail.
6. Le CFA scanne contrat de travail.
7. Le CFA téléverse la contrat de travail sur le système.
8. La scolarité FA valide le contrat de travail.
9. La scolarité FA vérifie le paiement des frais d'inscription.
10. La scolarité FA valide l'inscription.
11. La scolarité FA imprime la carte d'étudiant.

Flux alternatifs

- 7.a Le fichier téléversé est trop lourd.
 - 7.a.1 Le système alerte le CFA
 - 7.a.2 Le cas continue à l'étape 6.
- 8.a Le contrat n'est pas jugé conforme par la scolarité FA.
 - 8.a.1 La scolarité FA propose des amendement.
 - 8.a.2 Le cas continue à l'étape 5.
- 8.b Le contrat n'est pas signé.
 - 8.b.1 La scolarité FA le marque comme incomplet.
 - 8.b.2 Le CFA fait signer le contrat.
 - 8.b.3 Le cas continue à l'étape 6.

Cas référencés

- 9. Vérifier paiement
- 11. Imprimer carte étudiant

- ▶ En général, les sous-cas que l'on **étend** sont tout de même détaillés : les étapes du cas étendu sont souvent réparties au sein du cas étendu. Exemple : les cas 1, 2, 3, 9, 10 et 11 forment le descriptif du cas **Inscrire**.
- ▶ À l'inverse, les cas **inclus** le sont « en bloc », et peuvent donc être simplement référencés. Exemple : les cas 9 (inclue **Vérifier paiement**) et 11 (inclue **Imprimer carte étudiant**).
- ▶ On n'a pas de flux alternatif « 9.a Le paiement n'est pas valide » car ce cas est **exclu par la précondition**.

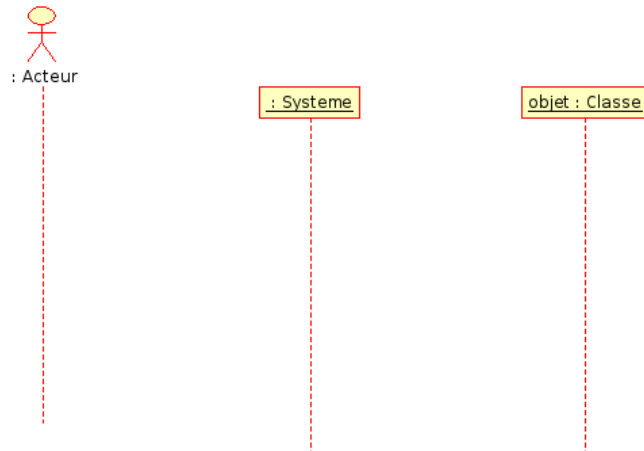
- ▶ Équilibre entre **détails** et **précision** : en général, au plus une dizaine d'étapes.
- ▶ Utiliser les **préconditions** à bon escient afin de limiter les flux alternatifs.
- ▶ Penser à faire **tous** les scénarios relatifs à un cas d'utilisation : ne pas oublier les **cas d'erreur** !

- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
 - Syntaxe (révisions IS2)
 - Diagrammes de séquence « système » ?
- 4 Diagramme d'activités (si on a le temps)

- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
 - Syntaxe (révisions IS2)
 - Diagrammes de séquence « système » ?
- 4 Diagramme d'activités (si on a le temps)

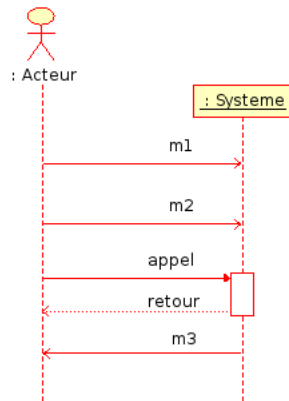
- ▶ On aime les diagrammes car ils sont **moins ambigus** que le texte.
- ▶ On aime les diagrammes car ils sont **plus concis** que le texte.
- ▶ On veut pouvoir **sous-spécifier** l'ordre des actions (↔ **concurrency**).

Des intervenants représentés par leur ligne de vie. qui s'échangent des messages ou appellent des opérations d'autres intervenants.



La différence entre **message** et **appel**.

- ▶ Lors d'un **message**, on ne se préoccupe pas de savoir si son interlocuteur est prêt. Pour un appel, il faut que l'interlocuteur soit disponible (**un seul appel à la fois**).
- ▶ Après l'envoi d'un message, je ne **suspends pas mon exécution**, pour un appel, je ne fais rien tant que je n'ai pas de **réponse**.
- ▶ Avec des messages, je n'ai pas de garantie sur l'**ordre** des événements : un message envoyé après peut arriver avant.



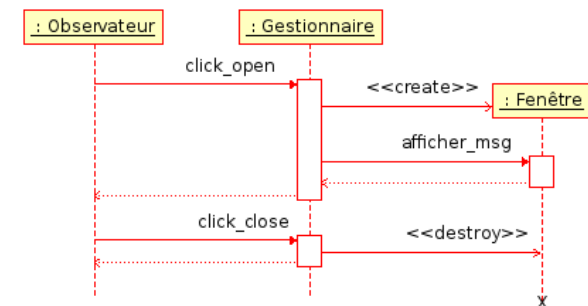
Ordres possibles ('!' = envoi, '?' = réception)

- ▶ *m1! m1? m2! m2? appel retour m3! m3?*
- ▶ *m1! m2! m1? m2? appel retour m3! m3?*
- ▶ *m1! m2! m2? m1? appel retour m3! m3?*
- ▶ *m1! m2! appel retour m1? m3! m3? m2?*
- ▶ ...

On sait seulement que

- ▶ L'envoi se fait avant la réception
- ▶ Sur une même ligne de vie, l'ordre des **messages sortants** est respecté.

- ▶ Messages qui créent ou détruisent des **objets**.
- ↪ On ne crée ni ne détruit un acteur ou un système!
- ▶ La création pointe (en général) sur l'objet et non sur sa ligne de vie.
- ▶ La destruction termine la ligne de vie de l'objet : plus d'interaction ultérieure.



Représentés par une boîte englobant une partie des interactions.

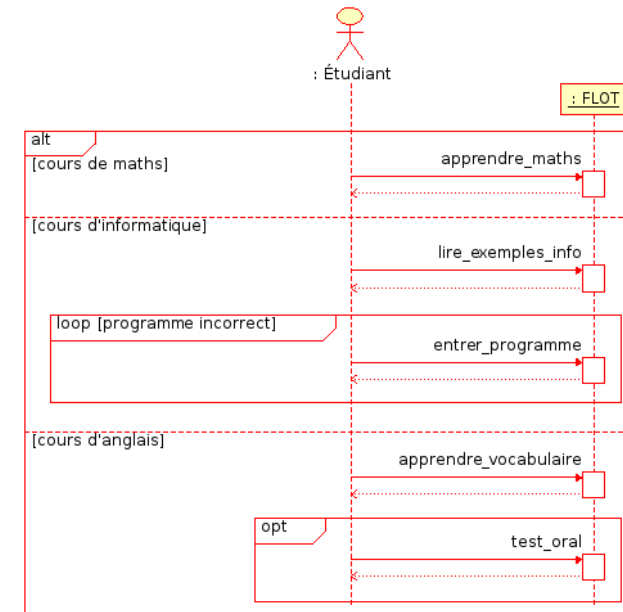
Ils permettent d'exprimer des flots de contrôle sur le diagramme :

opt Fragment optionnel, équivalent à un **if... then**.

alt Fragment d'alternative, équivalent à un **if ... then ... else**, voire à un **switch ... case**

loop Fragment de boucle, équivalent à un **while** ou un **for**.

Les **conditions** (plus ou moins précises) sont placées **en note**, **entre crochets** dans la partie idoine du fragment.

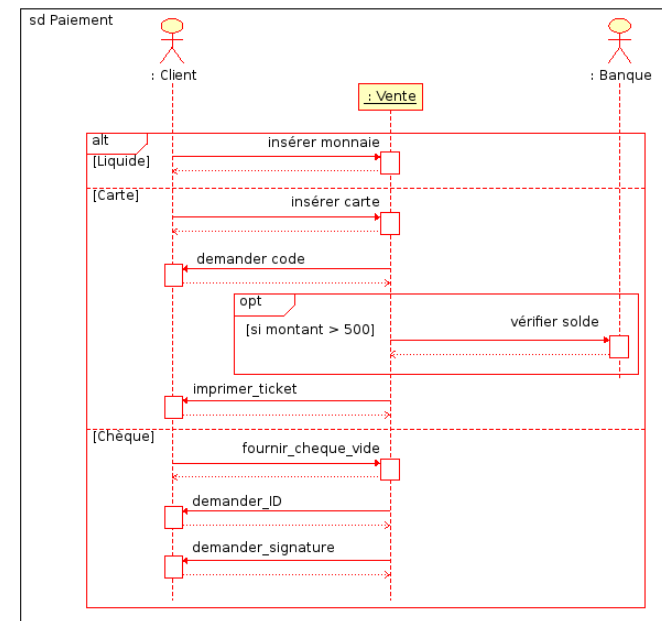


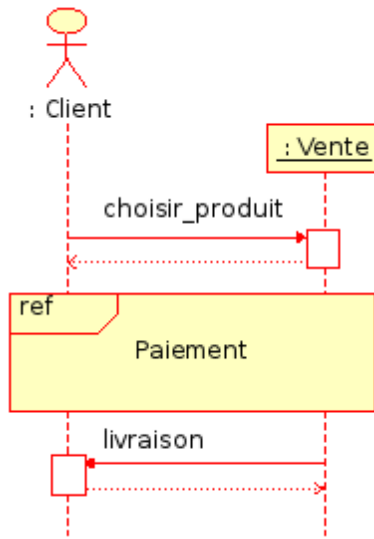
Pour les **références** : utile lorsqu'un scénario **inclut** un autre, ou lorsqu'on a déjà fait le diagramme d'une fonction annexe appelée.

sd Pour **sequence diagram**, englobe tout le diagramme afin de lui donner un nom.

ref Pour **reference**, agit au sein d'un diagramme comme si le diagramme référencé était présent.

À savoir lire et comprendre, et à savoir utiliser (cahier des charges...).





Pour la **concurrency** : pour limiter le nombre d'ordres possibles ou bien spécifier que plusieurs choses peuvent se faire en parallèle.

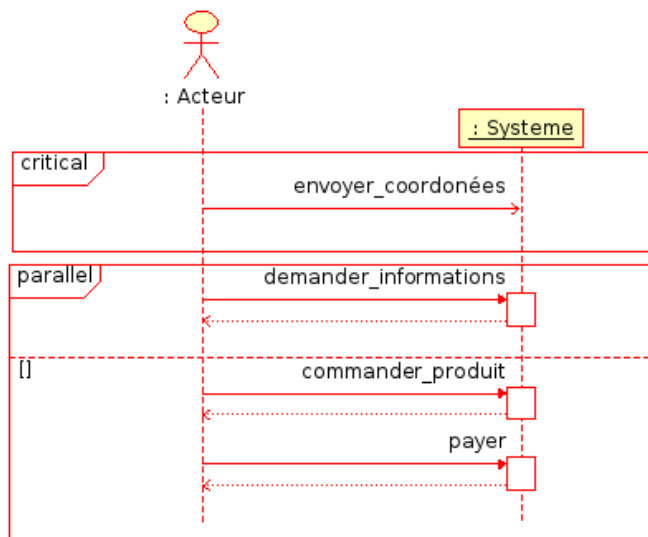
par L'ordre des événements sortants n'a plus d'importance : ils peuvent se faire en **parallèle**.

seq Chaque sous fragment doit être terminé avant de passer au suivant : ils se produisent en **séquence**.

critical Cette partie doit être traitée comme une portion **critique** : tout d'un bloc. Une séquence est donc équivalente à une suite de portions critiques.

À savoir lire et comprendre.

Les diagrammes de séquence ne sont pas faits pour spécifier la concurrence : les diagrammes d'activités sont plus adaptés.



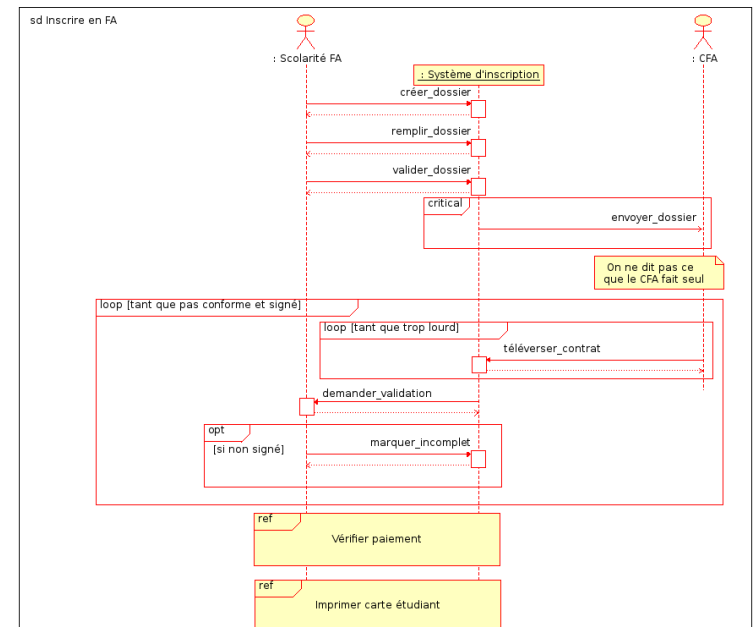
- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
 - Syntaxe (révisions IS2)
 - Diagrammes de séquence « système » ?
- 4 Diagramme d'activités (si on a le temps)

- ▶ On explicite le comportement du système au sein de son environnement.
- ▶ On ne montre pas ce que fait le système en interne.
- ▶ Le système est donc représenté par **une seule** ligne de vie.
- ▶ À l'inverse, dans un diagramme de séquence « interne », les **acteurs extérieurs n'apparaissent pas**.
- ▶ Pour rappel : dans un diagramme de séquence **système**, il n'y a ni création ni destruction.

- ▶ Les scénarios sont textuels :
 - ils sont potentiellement **ambigus** 😞
 - ils sont **compréhensibles** par des non-informaticiens (par exemple, le client) 😊
- ▶ Les scénarios décrivent (en partie) les actions des acteurs **qui ne sont pas des interactions** avec le système 😞
- ▶ Les scénarios fournissent aisément des **cas de test** 😊
- ▶ Les DSS fournissent quelques informations sur la **concurrency** 😊
- ▶ Les DSS permettent de modéliser des interactions complexes (**boucles, conditions...**) 😊
- ▶ Les DSS ne donnent que peu d'information sur les **moyens** d'interaction entre les acteurs et le système 😞

Caveat emptor
Ce n'est pas toujours possible/aisé à cause de la structure en goto des scénarios.

1. Placer les lignes de vie en fonction des acteurs.
2. Identifier qui initialise chaque action et qui en est le destinataire.
3. Pour chaque interaction, se poser la question de la concurrence (**message** ou **appel**?).
4. Structurer les flux alternatifs à l'aide de **conditions** et de **boucles**.
5. Ne pas détailler les cas référencés, mais utiliser les **ref**.
6. Nommer le diagramme
7. Faire les diagrammes des cas référencés.



Caveat emptor

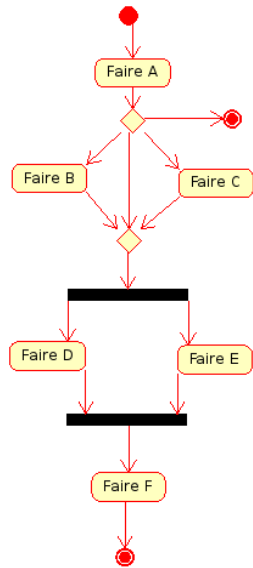
Ce n'est pas toujours possible/aisé à cause de la structure en goto des scénarios.

1. Placer les lignes de vie en fonction des acteurs.
2. Identifier qui initialise chaque action et qui en est le destinataire.
3. Pour chaque interaction, se poser la question de la concurrence (**message** ou **appel** ?).
4. Structurer les flux alternatifs à l'aide de **conditions** et de **boucles**.
5. Ne pas détailler les cas référencés, mais utiliser les **ref**.
6. Nommer le diagramme
7. Faire les diagrammes des cas référencés.

- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)
 - Syntaxe (révisions IS2)
 - Compliquons : ajoutons des objets

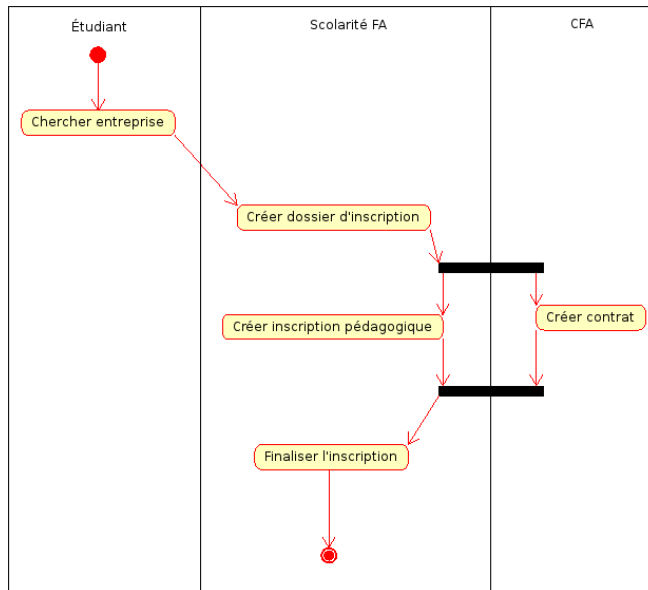
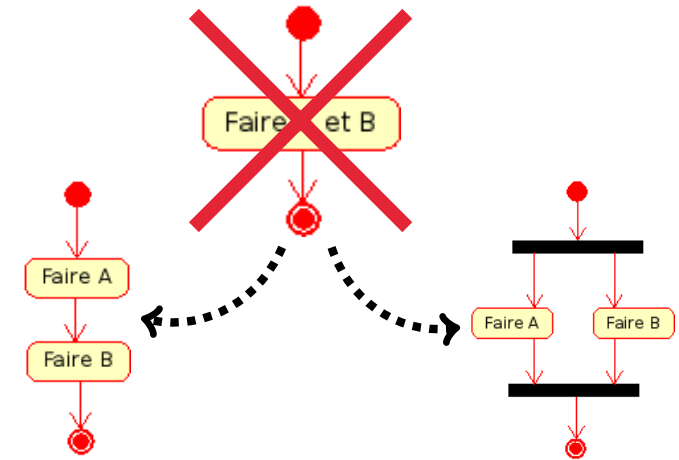
- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)
 - Syntaxe (révisions IS2)
 - Compliquons : ajoutons des objets

- ▶ Modéliser les **actions internes** des différent participants (acteurs, composants d'un système).
- ▶ Fort accent sur la **concurrence** : en **parallèle** ou en **séquence** ?
- ▶ On se préoccupe de qui fait quoi.
- ▶ Les interactions se font à travers l'**échange d'objets**.
- ▶ Modélisation **hiérarchique**.

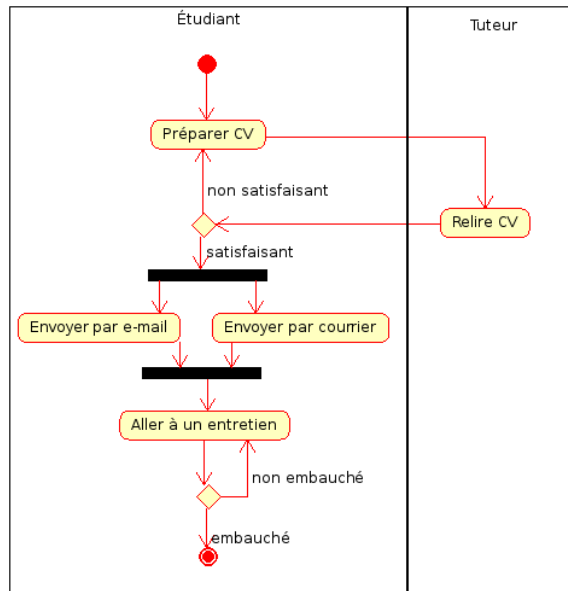


- ▶ Départ du diagramme : unique.
- ▶ Activités : verbe à l'infinif + complément.
- ▶ Choix (losange) : activé si **une** entrée ont terminé ; **une** sortie est choisie : **OU**.
- ▶ Parallélisation/synchronisation (barre) : activé si **toutes** les entrées ont terminé ; **toutes** les sorties sont prises : **ET**.
- ▶ Fin du diagramme : potentiellement plusieurs.

Pas de « et » dans les intitulés d'activités !



- ▶ Une activité peut elle-même être décomposée.
- ▶ En entrant dans l'activité, on entre dans le point d'entrée du diagramme d'activité correspondant.
- ▶ En passant par n'importe quel point de sortie, on a terminé l'activité.



- 1 Programme This is so meta !
- 2 Scénarios textuels
- 3 Diagrammes de séquence
- 4 Diagramme d'activités (si on a le temps)
 - Syntaxe (révisions IS2)
 - Compliquons : ajoutons des objets

- ▶ Pour indiquer lorsqu'il y a **interaction** à travers un même objet.
- ▶ Pour indiquer les **changements d'états** des objets induits par l'activité.
- ▶ Syntaxe : boîte entre deux activités ; l'état de l'objet est précisé.

