

Méthodes pour l'informatisation - compléments

Modélisation des interactions

Christine Plumejeaud

Doctorante Informatique UJF

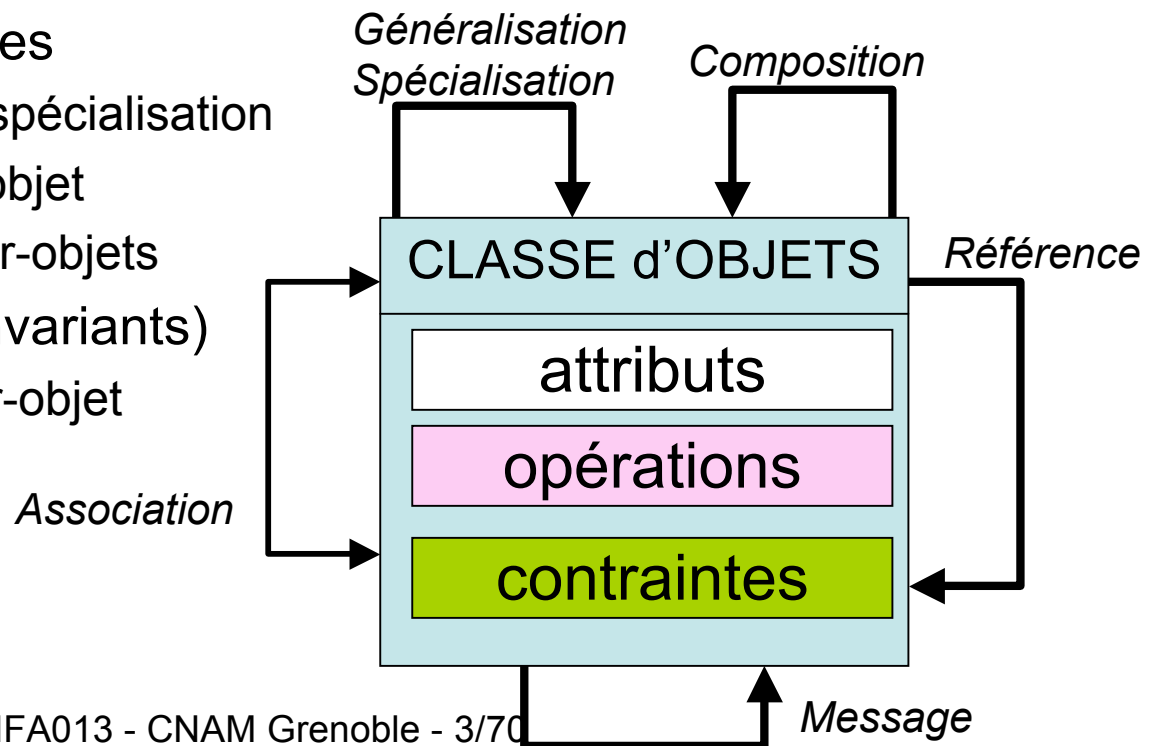
CNAM - centre de Grenoble

Plan

1. Introduction
2. Capture des besoins : les cas d'utilisation
3. Cas d'étude
4. Diagramme d'activité
5. Diagramme de séquence

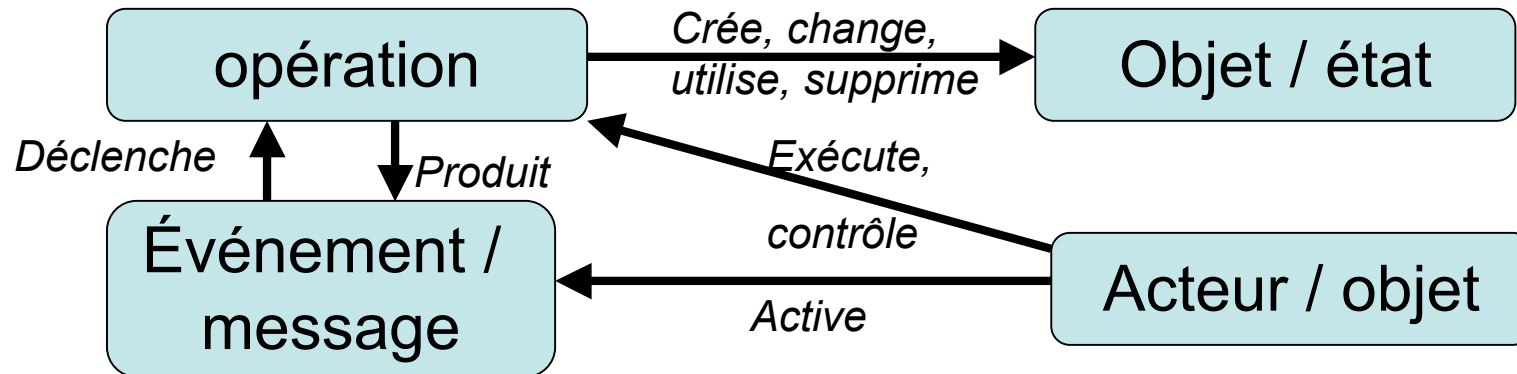
Aspects statiques

- Identification de l'objet
- Valeurs de l'objet
- Structuration complexe des données : produit cartésien, ensemble, liste
- Graphes de classes
 - Généralisation/spécialisation
 - Composition d'objet
 - Association inter-objets
- Contraintes (ou invariants)
 - Intra-objet, Inter-objet



Aspects dynamiques

- Donner une vie à des objets qui naissent, évoluent et meurent
- États pour contrôler l'évolution
- Evènements pour déclencher des changements d'états
- Messages pour supporter des évènements
- Opérations pour gérer les objets en réaction aux évènements
- Acteurs pour contrôler et exécuter les opérations
- Flux d'information pour communiquer
- Architecture des applications



Les diagrammes UML

1. Besoin des utilisateurs : cas d'utilisation
2. Aspects statiques
 - diagrammes de classes
 - diagrammes d'objets
3. Aspects dynamiques
 - diagrammes de séquence,
 - diagrammes de collaboration,
 - diagrammes d'activités
 - automates d'états-transitions
4. Aspects physiques
 - Les diagrammes de composants
 - Les diagrammes de déploiement

Les trois modèles

1. Le modèle de classes

- -aspects statiques et structurels, orienté « données »

2. Le modèle d'états

- Aspects temporels, comportementaux et de « contrôle » du système

3. Le modèle d'interactions

- Collaborations entre les différents objets, les aspects associés aux « interactions » du système

Les trois modèles peuvent se lire indépendamment les uns des autres, mais *complémentaires*, et offrent ensemble une vue aussi complète que possible du système.

Le modèle d'interactions

Décrit les interactions entre objets, et la façon dont ils collaborent --> comportement global du système

1. Cas d'utilisation

- capturent les besoins, et les interactions du système avec l'extérieur

2. Les diagrammes de séquence

- représentent la séquence des interactions entre les objets dans le temps

3. Les diagrammes d'activité

- Illustrent le flux de contrôle entre les étapes d'un traitement

Capturer les besoins des utilisateurs

1. Un SI doit répondre aux besoins des utilisateurs. Leur capture nécessite :
 - l'étude du système existant
 - l'acquisition des nouveaux besoins
 - Fonctionnels : que doit faire le futur SI ?
 - Non fonctionnels : performances, sécurité, etc.
 - Utilisabilité : contexte de l'usage
 - L'utilisation de techniques d'acquisition
 - Analyser la documentation existante
 - Questionner, observer les utilisateurs

Décrire les besoins des utilisateurs

Spécifier avec UML avec les diagrammes de cas d'utilisation pour modéliser et documenter les besoins

- Un cas d'utilisation n'est pas un besoin, mais une fonctionnalité du système devant répondre à un besoin
- Les diagrammes de cas d'utilisation ne permettent pas de répondre à des besoins non fonctionnels
- Il est nécessaire d'établir la liste des besoins non-fonctionnels pour compléter la spécification des besoins

Diagramme de cas d'utilisation (*use case*)

1. Formalisé par Ivar Jacobson
2. Décrit sous la forme d'actions et de réactions le comportement d'un système du *point de vue utilisateur*
3. Définissent les *limites du système* et les relations entre celui-ci et son environnement.
4. Manière spécifique d'utiliser un système : image d'une fonctionnalité du système déclenchée en réponse à la *stimulation d'un acteur externe*.

Exemple de besoin

Gestion des malades dans un cabinet médical


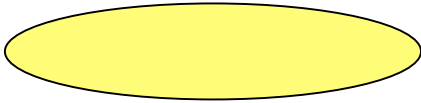
1. Un médecin doit pouvoir

- Créer des dossiers médicaux des patients, les modifier, consulter, supprimer.
- Créer des ordonnances, modifier, imprimer et supprimer
- Créer et modifier des instructions à l'attention du secrétariat
- Créer et modifier les informations sur le patient (informations non médicales)

2. Une secrétaire doit pouvoir

- Créer et modifier des informations sur un patient (si informations non médicales)
- Editer une feuille de soin
- Imprimé un récépissé lors de l'encaissement d'un paiement
- Etc.

Diagramme d'utilisation : concepts

- Il comprend les acteurs, le système et les cas d'utilisation eux-mêmes.
- Les acteurs  déclenchent des cas d'utilisation 
- les cas d'utilisation sont contenus dans le système

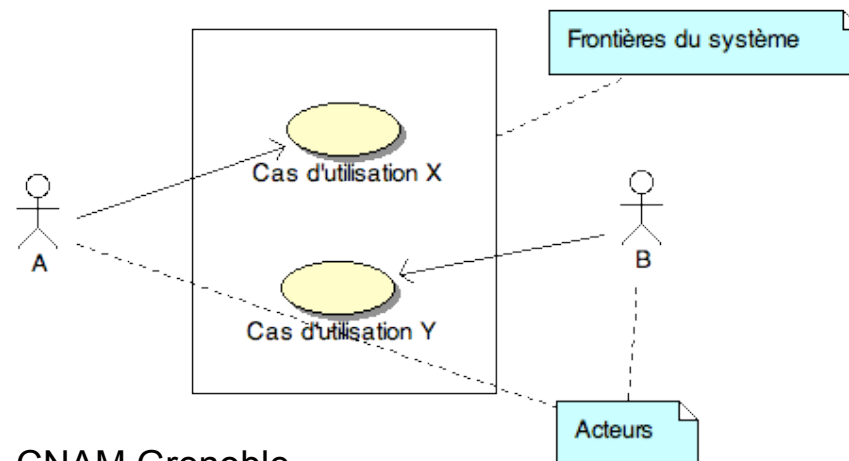


Diagramme d'utilisation : l'acteur

1. Un acteur représente tout ce qui est *externe au système*, humain ou non, qui *interagit* avec le système et qui correspond à une catégorie d'utilisateurs (plus précisément à un *rôle*).
2. Les acteurs se déterminent en observant les utilisateurs directs du système ainsi que les autres systèmes interagissant avec le système en question
3. La même personne physique peut jouer le rôle de plusieurs acteurs (vendeur, client). Plusieurs personnes peuvent jouer le même rôle et donc agir comme le même acteur (tous les clients). Le nom de l'acteur décrit le rôle joué par l'acteur.

Diagramme d'utilisation : l'acteur

Dans le cas du cabinet médical, nous avons
2 acteurs :

- le médecin
- la secrétaire

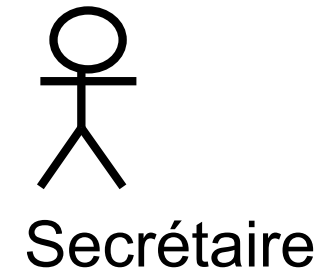
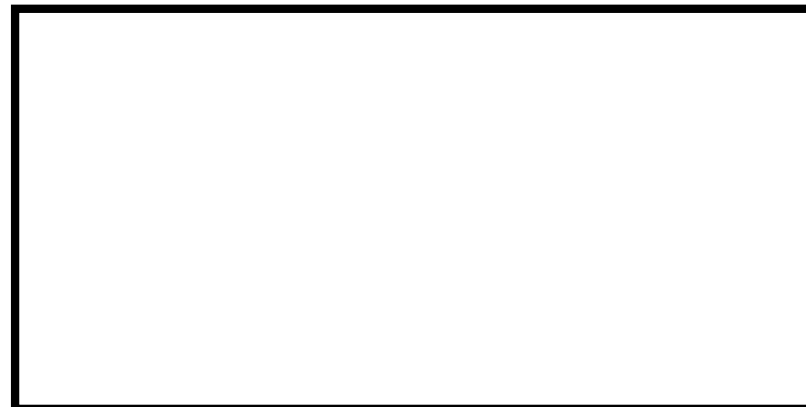
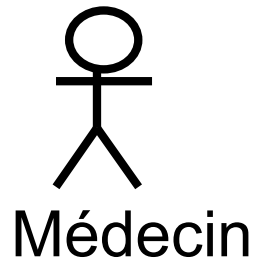


Diagramme d'utilisation : les cas d'utilisation

Un cas d'utilisation

- représenté graphiquement par une ellipse avec le nom du cas en dessous
- décrit une séquence d'action effectuée par le système pour livrer un résultat à l'acteur

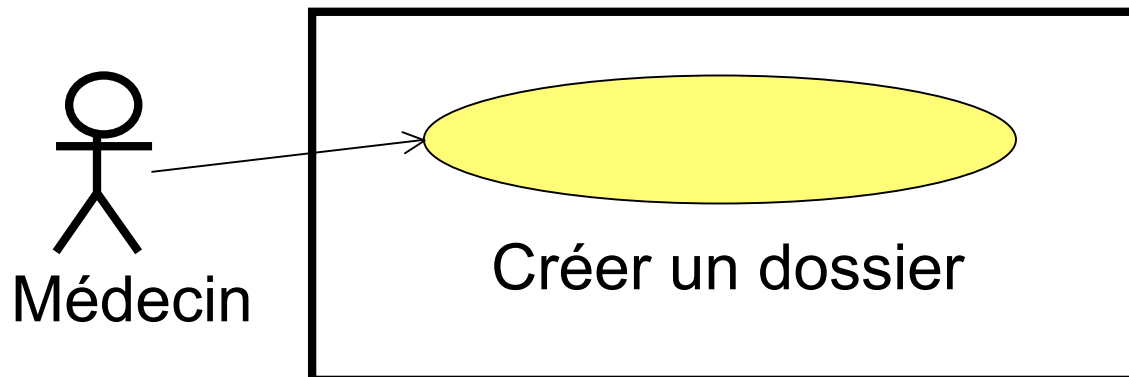


Diagramme d'utilisation : les cas d'utilisation

- Les cas d'utilisation se déterminent en observant et en précisant, acteur par acteur, les séquences d'interaction - les scénarios - du point de vue des l'utilisateur.
- La participation de l'acteur est signalée par un lien de communication entre l'acteur et le cas d'utilisation. Ce lien peut être orienté pour signaler l'initiateur de l'interaction.

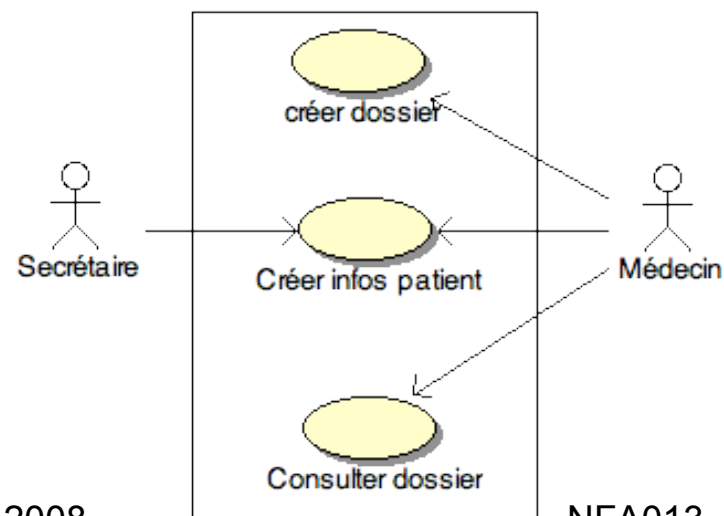
Diagramme d'utilisation : les cas d'utilisation

1. Les cas d'utilisations doivent être vus comme des classes dont les instances sont des *scénarios*. Chaque fois qu'un acteur interagit avec le système, le cas d'utilisation instancie un scénario; ce scénario correspond au flot de messages échangés par les objets durant cette interaction.
2. Les cas d'utilisation servent de fil conducteur pour l'ensemble du projet

Le modèle de cas d'utilisation

Le modèle de cas d'utilisation

- Comprend une collection de cas d'utilisation
- Caractérise le comportement de l'ensemble du système et des acteurs externes dans leurs interactions.

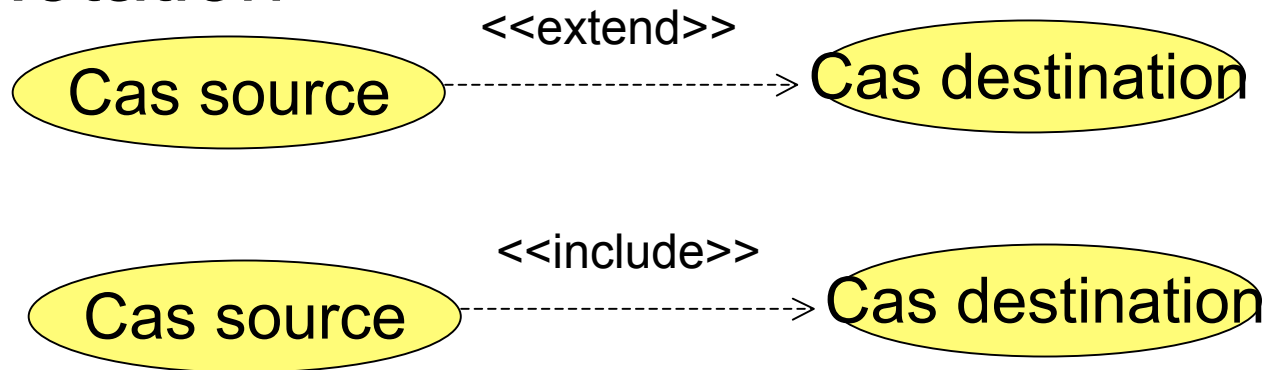


Raffinement des cas d'utilisation

Relations entre cas d'utilisation

- Extend : stéorotype <<extend>>
- Include : stéorotype <<include>>

Notation



La relation « *include* »

L'inclusion entre 2 cas d'utilisation signifie que toute instance du cas source comprend nécessairement le cas destination.

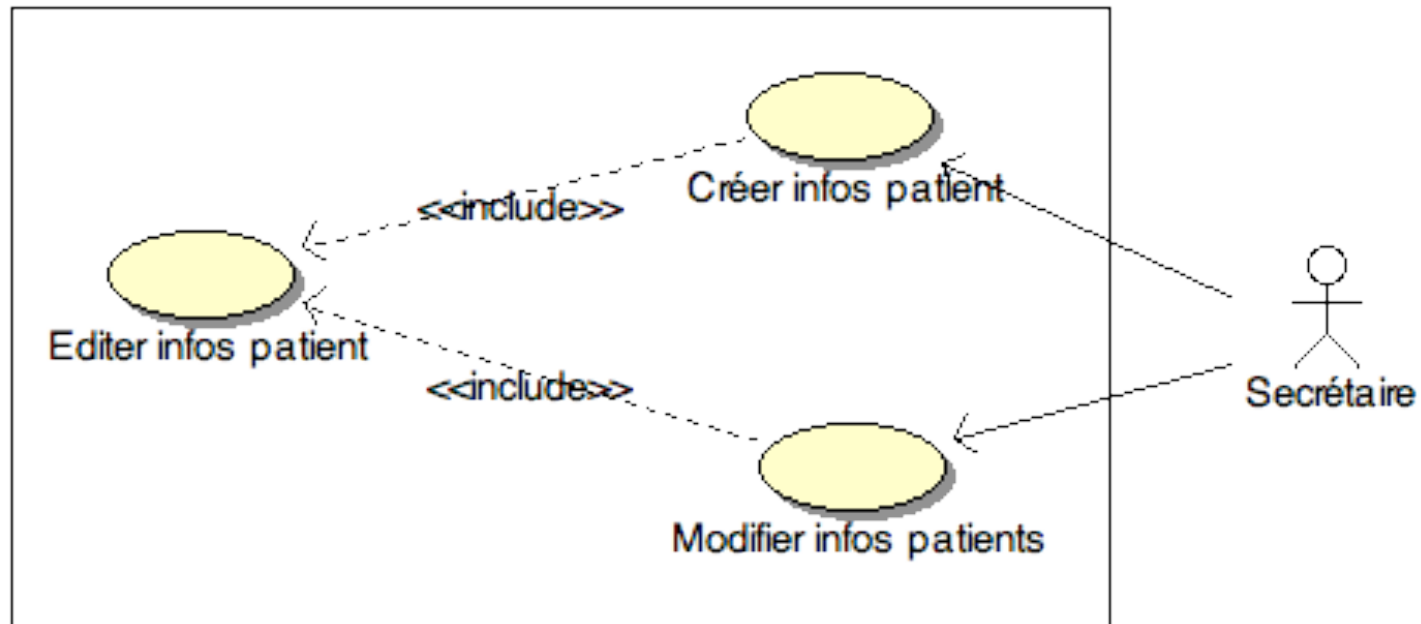
Quand l'utiliser ?

Lorsqu'on veut réutiliser un ensemble d'interactions communes dans plusieurs cas d'utilisation : on regroupe alors les interactions communes dans un cas d'utilisation (destination), lié aux cas d'utilisation qui l'emploient (les cas sources)

Avantage : réutilisation

La relation « *include* »

Exemple



La relation « *extend* »

L'extension entre 2 cas d'utilisation signifie que le cas source étend le comportement du cas destination

Quand l'utiliser ?

- Lorsque l'on veut enrichir un cas de base avec une possibilité de fonctionnement non décrit dans le cas source.
- Cette extension est décrite dans un cas à part, et n'a pas de sens SEULE (sans son cas source).
- Les 2 cas sont ensuite reliées par la relation d'extension

Les points d'extension montrent à quel moment survient l'extension

Une condition peut-être rajoutée à la relation d'extension pour la préciser.

La relation « *extend* »

Exemple : pompe à essence

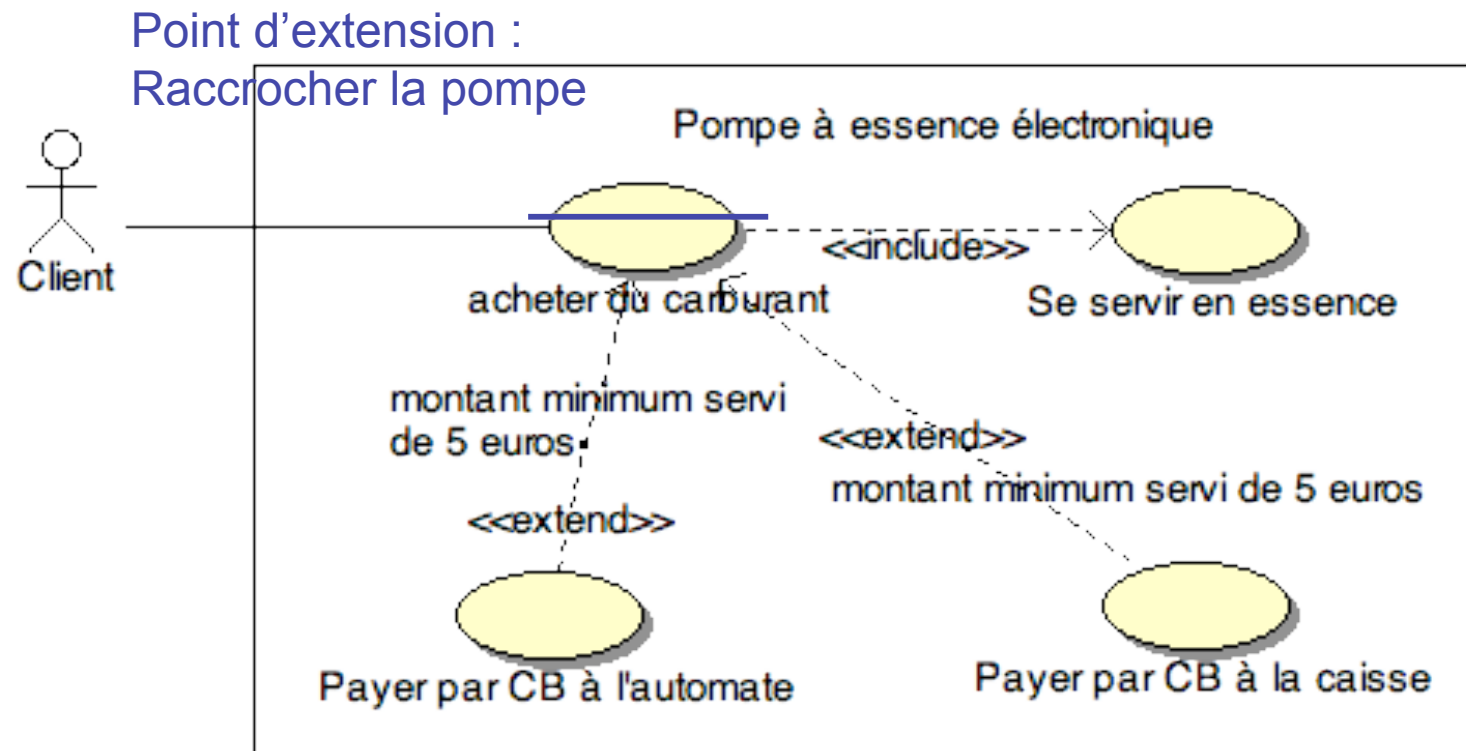


Diagramme d'utilisation : les cas d'utilisation

- Un cas d'utilisation
 - Est un ensemble complet d'actions (avec des événements de début et de fin, les acteurs impliqués dans les actions) et de règles qui régissent l'enchaînement des actions.
 - Il définit les interactions entre le système et les acteurs
 - Il inclut le déroulement normal et tous les déroulements alternatifs
- Un scénario
 - Est un *déroulement spécifique* des événements. Ce déroulement dépend des événements à l'origine et à l'issue de chaque action spécifiée dans le cas d'utilisation et dont dépend l'enchaînement des actions.
 - Il est possible de dériver plusieurs scénarios à partir d'un cas d'utilisation. Il suffit pour cela que l'enchaînement des actions ne soit pas une simple séquence.

Description des cas d'utilisation :

La description peut-être sous une forme textuelle et peu structurée

Exemple :

« le médecin cherche le patient dans la liste des patients. Si celui-ci n'existe pas il le crée, sinon il crée son dossier. Le médecin introduit les informations sur les antécédents du patient, ses allergies, la liste des substances auxquelles il est allergique, la liste des traitements qu'il suit et la durée de ces traitements. Etc. »

Description des cas d'utilisation

Elle peut être décomposée en précisant les interactions entre le système et l'acteur en distinguant le déroulement de base des déroulements alternatifs.

Créer ordonnance

Acteur	Réponse du système
1. Le médecin demande à créer une ordonnance 2. Le médecin choisit le patient souhaité 3. Le médecin sélectionne un médicament dans une liste ...	2. Le système lui demande de choisir un patient 4. Le système édite une ordonnance vierge 6. Le système demande à saisir les doses et la fréquence des prises
Déroulement alternatif 5-6 Le médecin entre le nom du médicament, et la posologie	

Description des cas d'utilisation :

La description d'un cas d'utilisation comprend les éléments suivants :

- Le *début* : « le cas débute quand X se produit »
- La *fin* : « le cas se termine quand Y s'est produit »
- *L'interaction* entre le cas et les acteurs
- Les échanges *d'informations* : par exemple, « l'utilisateur se connecte au système et donne son nom et son mot de passe »
- La *chronologie* et l'origine des informations
- Les *répétitions de comportement* qui peuvent être décrites au moyen de pseudo-code, avec des constructions du type :

Boucle
...faire quelque chose
Fin de boucle

Tant que
...faire quelque chose
Fin tant que

Règles de mise en œuvre des cas d'utilisation

- Un cas d'utilisation décrit une fonctionnalité ou une motivation et aussi une interaction entre un acteur et un système sous la forme d'un flot d'évènements
- La description de l'interaction se concentre sur ce qui doit être fait
- Un cas d'utilisation doit être simple
- Il ne faut pas mélanger les cas d'utilisation
- Un cas d'utilisation doit éviter d'employer des expressions floues et imprécises.

Règles de mise en œuvre des cas d'utilisation

Il est primordial de trouver le *bon niveau d'abstraction*. Les réponses apportées aux 2 interrogations suivantes peuvent servir de gabarit :

- Est-il possible d'exécuter une activité donnée indépendamment des autres, ou faut-il toujours l'enchaîner avec une autre activité :
 - « passer un appel téléphonique »,
 - « composer un numéro de téléphone »
- Est-il judicieux de regrouper certaines activités en vue de les documenter, tester ou modifier ?

Construction des cas d'utilisation

- En règle générale, il n'y a qu'un acteur par cas d'utilisation
- Lors de la construction, il faut se demander :
 - Quelles sont les tâches de l'acteur
 - Quelles informations l'acteur doit-il créer, sauvegarder, modifier, détruire ou lire ?
 - L'acteur devra-t-il informer le système des changements externes
 - Le système devra-t-il informer l'acteur des conditions internes
- Les cas d'utilisation peuvent :
 - être présentés au travers de vues multiples
 - être groupés selon leurs séquences de déclenchement types ou en fonction des différents points de vue

Processus d'élaboration des cas d'utilisation

- Définir un guide de style pour la rédaction
- Définir grossièrement les cas d'utilisation
- Approfondir la compréhension et la description d'un cas d'utilisation particulier. Identifier les scénarios
- Un scénario est un chemin particulier au travers de la description abstraite et générale fournie par le cas d'utilisation

Etude de cas

Guichet Automatique de Billets (GAB)

*Source : UML par la pratique,
P. Roques*

Enoncé simplifié

1. Le GAB offre les services suivants :

- Distribuer de l'argent à tout porteur de Carte Bancaire (CB) de type VISA ou de la banque
- Pour un client porteur d'une carte de la banque : consulter le solde du compte, déposer des espèces ou des chèques

2. Il faut savoir que :

- Toutes les transactions sont sécurisées
- Il est nécessaire d'alimenter le distributeur en argent, papier pour l'impression des tickets, récupérer les chèques déposés, les cartes avalées, et les espèces déposées.

Identifier les acteurs du GAB

1. Un acteur représente une catégorie d'individus externes au système et qui ont le même comportement vis à vis de celui-ci : attendre un service
2. Un acteur peut-être un agent humain ou un autre système
3. Un acteur est dit *primaire* lorsqu'il est l'initiateur du cas d'utilisation. Il est dit *secondaire* lorsque le déroulement du cas d'utilisation nécessite son intervention.

Identifier les acteurs du GAB

1. Acteurs principaux

- Porteur de CB visa
- Client de la banque
- Agent de maintenance

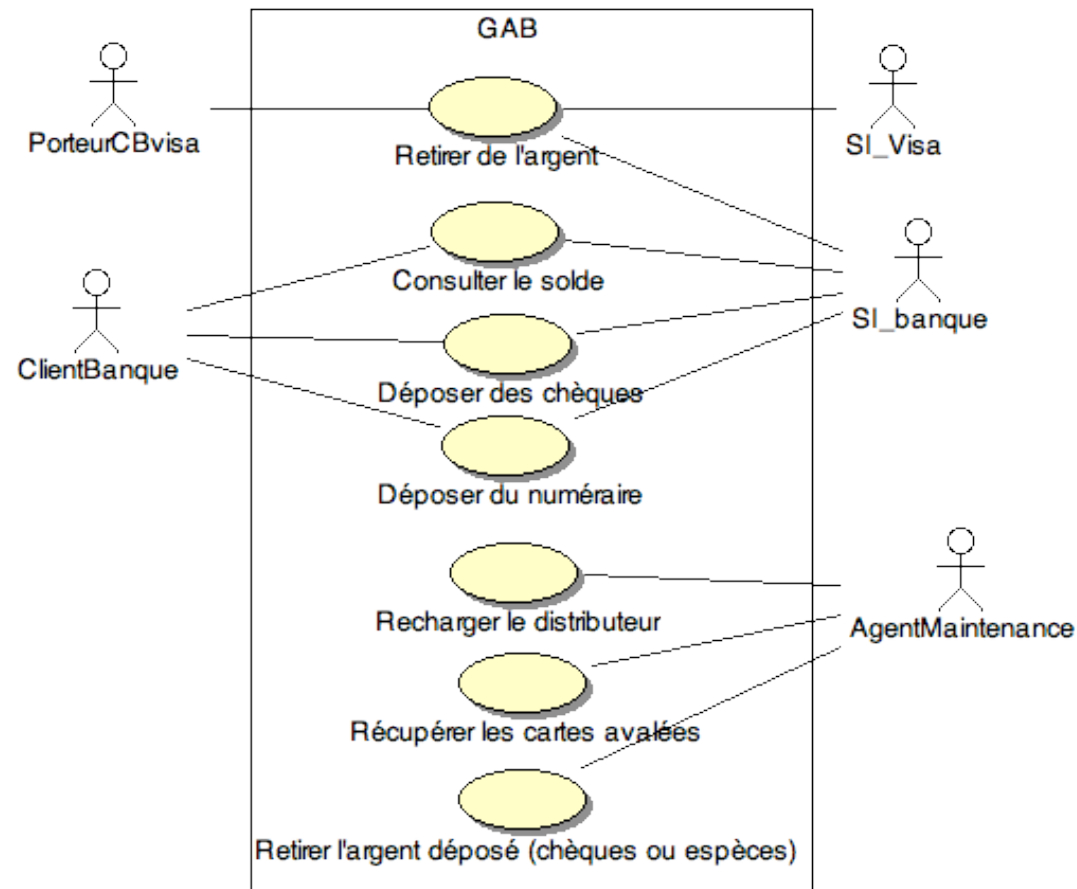
2. Acteurs secondaires

- SA visa (organisme de carte bancaire)
- SI banque (Système d'Information de la banque)

Identifier les cas d'utilisation

1. Pour un porteur de CB visa
 - Retirer de l'argent
2. Pour un client de la banque
 - Retirer de l'argent
 - Consulter le solde
 - Déposer des chèques
 - Déposer des numéraires
3. Pour l'agent de maintenance
 - Retirer des chèques et numéraires
 - Récupérer les cartes avalées
 - Recharger le distributeur

Diagramme de cas d'utilisation du GAB



Documenter les cas d'utilisation

- Un cas d'utilisation ne se limite pas à sa représentation graphique, sa documentation n'est pas optionnelle mais *nécessaire* et obligatoire pour la définition du cas
- La documentation doit expliciter les activités à dérouler, le déroulement *nominal* et le déroulement *exceptionnel*, les déroulements *alternatifs* (interruption du déroulement nominal du cas initial par un autre cas d'utilisation pour les besoins du cas initial) et les *pré et post conditions* pour s'assurer du bon déroulement du cas.

Documenter les cas d'utilisation

Titre : retirer de l'argent avec un carte visa

Résumé : ce cas permet au porteur de CB qui n'est pas un client de la banque de retirer de l'argent, si son crédit hebdomadaire le permet

Date de création : 02/02/2008

Date de mise à jour : 03/03/2008

Version : 2.2

Responsable : C. Plumejeaud

Pré conditions :

La caisse au GAB est alimentée

Aucune CB ne se trouve dans le lecteur

Documenter les cas d'utilisation

Scénario nominal

1. Le porteur de CB introduit la carte dans le lecteur du GAB
2. Le GAB vérifie que la carte est de type Visa
3. Le GAB demande au porteur de saisir son code d'identification
4. Le porteur saisit son code d'identification
5. Le GAB compare le code saisi avec celui de la puce de la carte
6. Le GAB demande une autorisation au SI_visa
7. SI_visa donne son accord en indiquant le solde hebdomadaire
8. Le GAB demande au porteur de CB d'indiquer le montant souhaité
9. Le porteur saisit le montant
10. Le GAB contrôle le montant demandé par rapport au solde hebdomadaire
11. Le GAB demande au porteur de CB s'il veut un ticket
12. Le porteur de CB demande un ticket
13. Le GAB rend la carte au porteur de CB
14. Le porteur de CB reprend sa carte
15. Le GAB délivre les billets et un ticket
16. Le porteur de CB prend les billets et le ticket.

Documenter les cas d'utilisation

Enchaînement alternatifs

A1 : code d'identification provisoirement erroné. L'enchaînement A1 commence au point 5 du scénario nominal

- Le GAB indique au porteur que le code est erroné, pour la première ou seconde fois
- Le GAB enregistre l'échec sur la carte

Le scénario reprend au point 3

A2 : montant demandé supérieur au solde hebdomadaire.

L'enchaînement A2 commence au point 10 du scénario nominal

- Le GAB indique que le montant est supérieur au solde hebdomadaire

Le scénario reprend au point 8

A3 : ticket refusé. L'enchaînement A3 commence au point 11 du scénario nominal

- Le porteur refuse les billets

Le scénario reprend au point 13

Documenter les cas d'utilisation

Enchaînement d'exception

E1 : carte non valide. L'enchaînement E1 commence au point 2 du scénario nominal

- Le GAB indique au porteur que la carte n'est pas valide et la confisque
- Le cas d'utilisation est terminé

E2 : le code d'identification est faux pour la 3eme fois. L'enchaînement E2 commence au point 5 du scénario nominal

- Le GAB indique au porteur que le code est erroné pour la troisième fois
 - Le GAB confisque la carte
 - Le SI_visa est informé
- Le cas d'utilisation est terminé

E3 : Retrait non autorisé. L'enchaînement E3 commence au point 6 du scénario nominal

- SI_visa interdit tout retrait
 - Le GAB éjecte la carte
- Le cas d'utilisation est terminé

Documenter les cas d'utilisation

Enchaînement d'exception (suite)

E4 : carte non reprise. L'enchaînement E4 commence au point 13 du scénario nominal

- Le GAB confisque la carte au bout de 15 secondes
- Le SI_visa est informé

Le cas d'utilisation est terminé

E5 : Billets non pris. L'enchaînement E5 commence au point 15 du scénario nominal

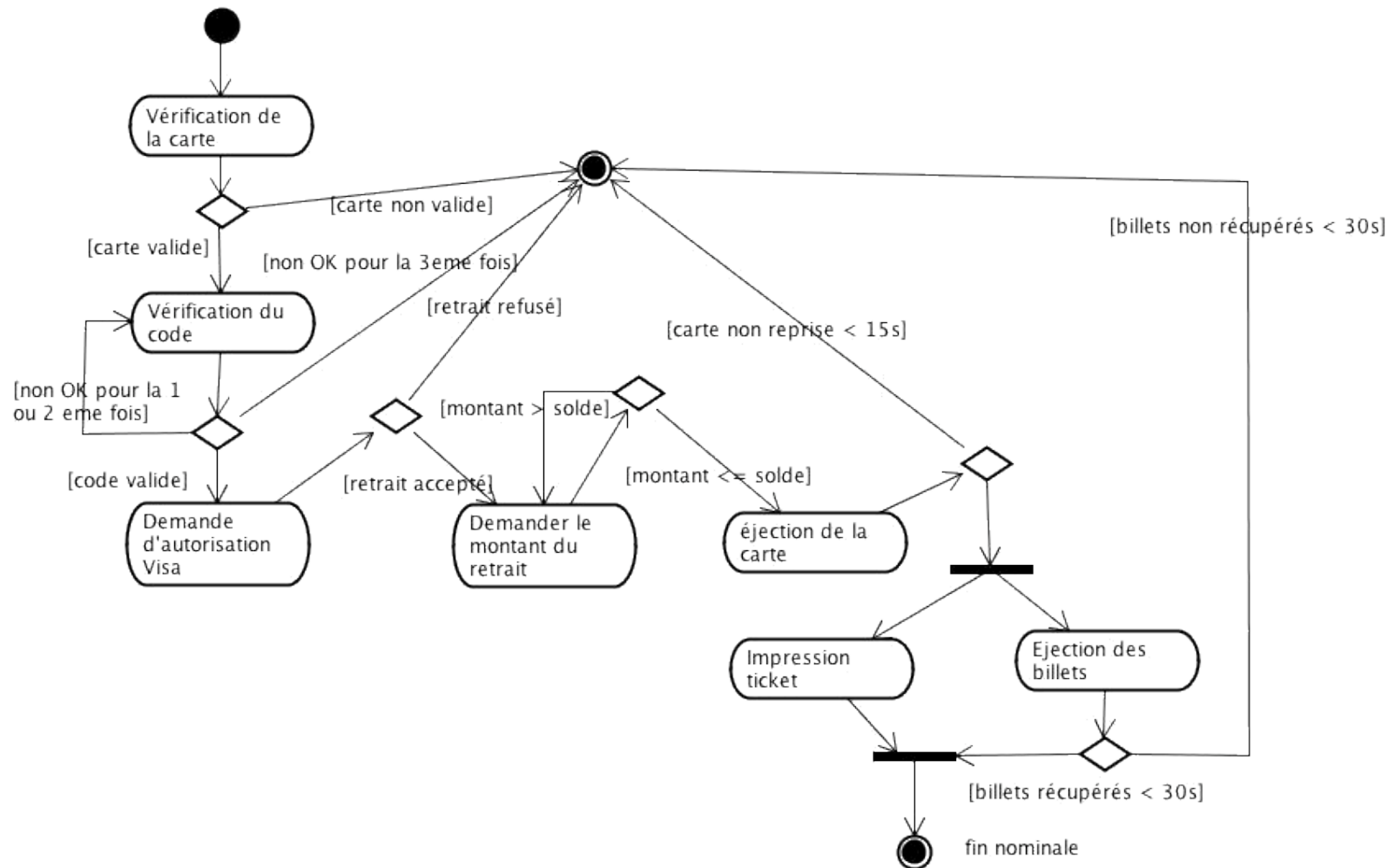
- Le GAB reprend les billets au bout de 30 secondes
- Le SI_visa est informé

Le cas d'utilisation est terminé

Postconditions :

La caisse du GAB est mise à jour en fonction du montant des retraits effectués.

Diagramme d'activité du GAB



Les diagrammes d'activité

Le diagramme d'activité est utilisé pour

- Modéliser une tâche (dans la modélisation métier)
- Décrire une fonctionnalité du système modélisée par un diagramme d'utilisation
- Décrire la logique d'une *opération* ou d'un algorithme.
- Décrire les activités et leur enchaînement dans un processus

Le diagramme d'activités : concepts

Diagramme d'activités =

- Ensemble d'activités liées par :
 - Transitions (séquentielle)
 - Transitions alternatives (conditionnelle)
 - Synchronisation (disjonction et conjonction d'activités)
- Deux états : initial et final
- Couloirs d'activité : représente le responsable de l'activité.

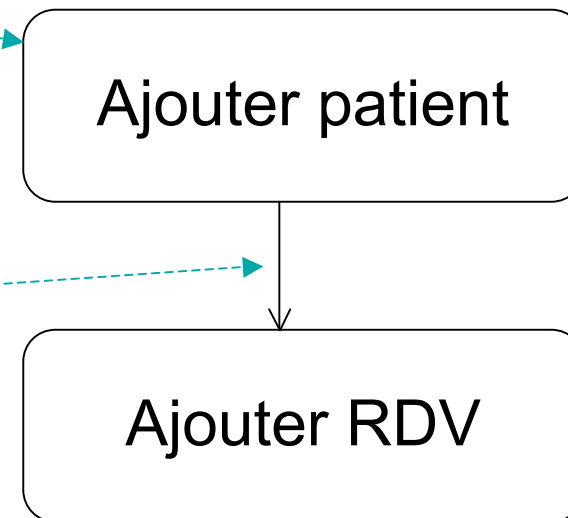
Notations du diagramme d'activité

- Activités

- Un rectangle arrondi
- Un nom significatif

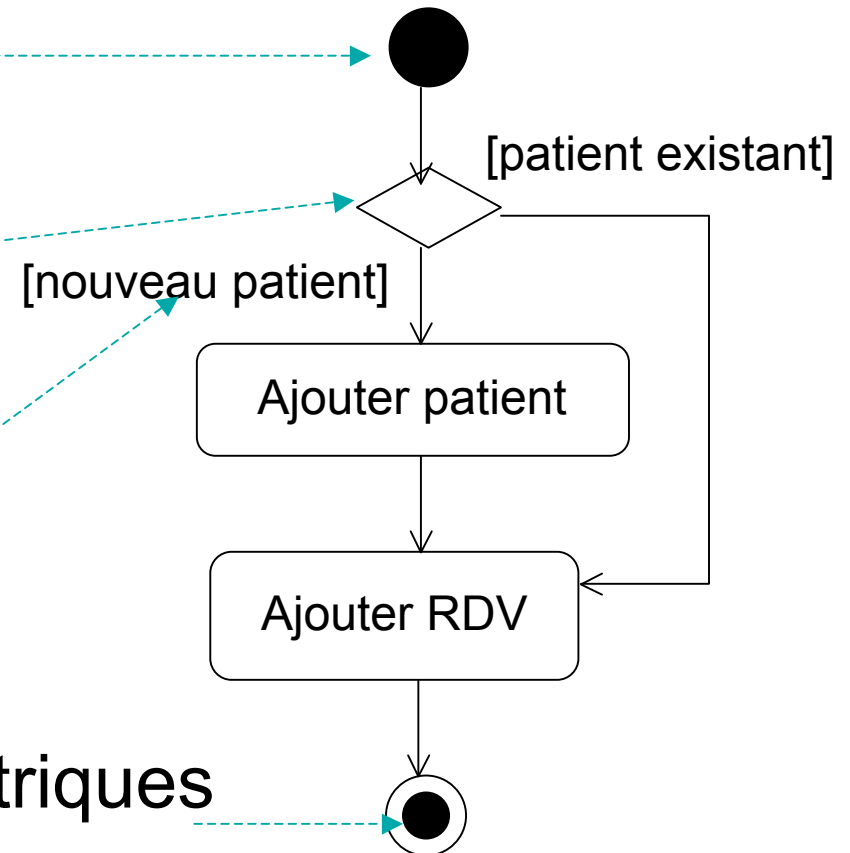
- Transitions

- flèche



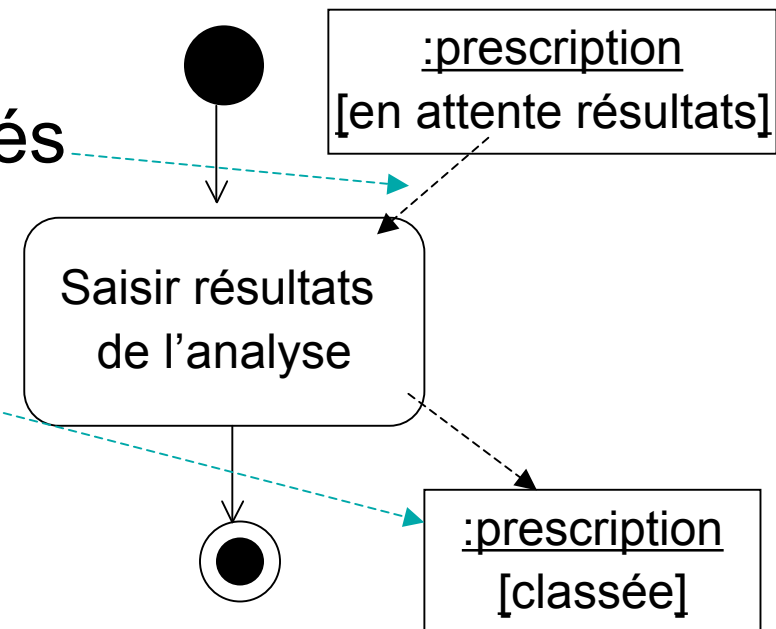
Notations du diagramme d'activité

- Etat initial
 - Cercle noirci
- Point de décision
 - Losange
- Condition (garde)
 - Entre crochets
- Etat final
 - Deux cerce concentriques dont un noir



Notations du diagramme d'activité

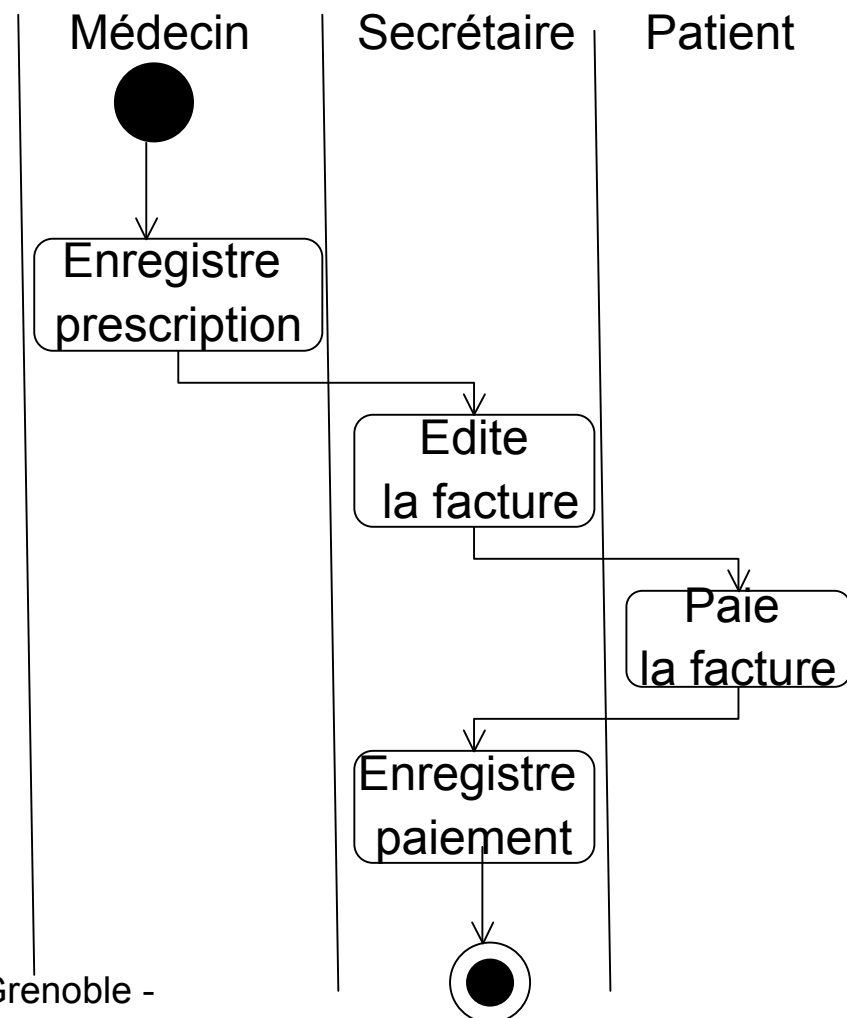
- Flot d'objets
 - Des flèches en pointillés
- Objets
 - Rectangle
 - Le nom de l'objet est souligné
 - On peut montrer l'état de l'objet (noté entre crochets)



Notations du diagramme d'activité

1. Couloirs d'activité

- Colonnes verticales
- Etiquetés des noms de personnes, département, etc., responsables de cette activité



Notations du diagramme d'activité

Fils d'exécutions concurrents

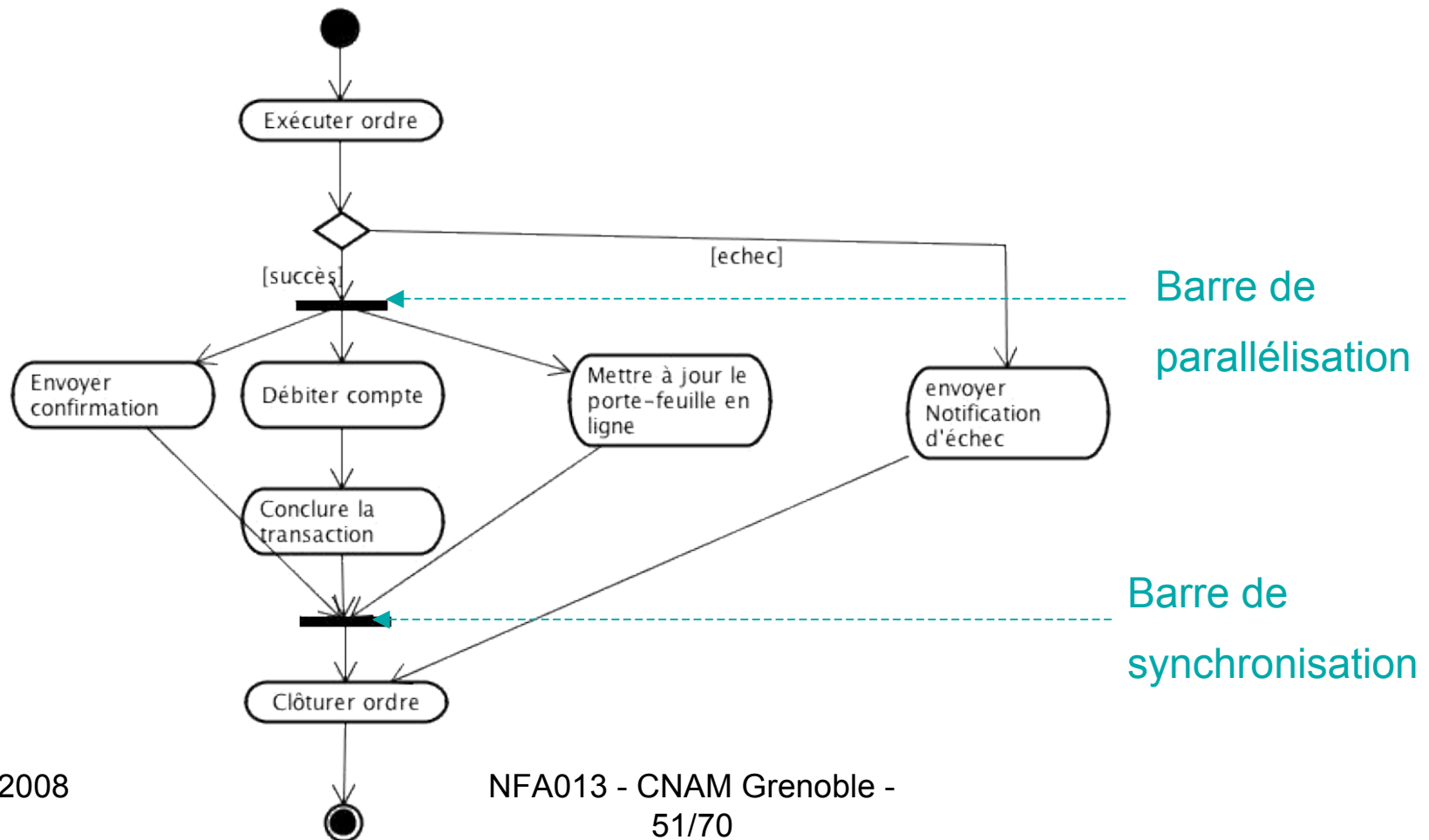
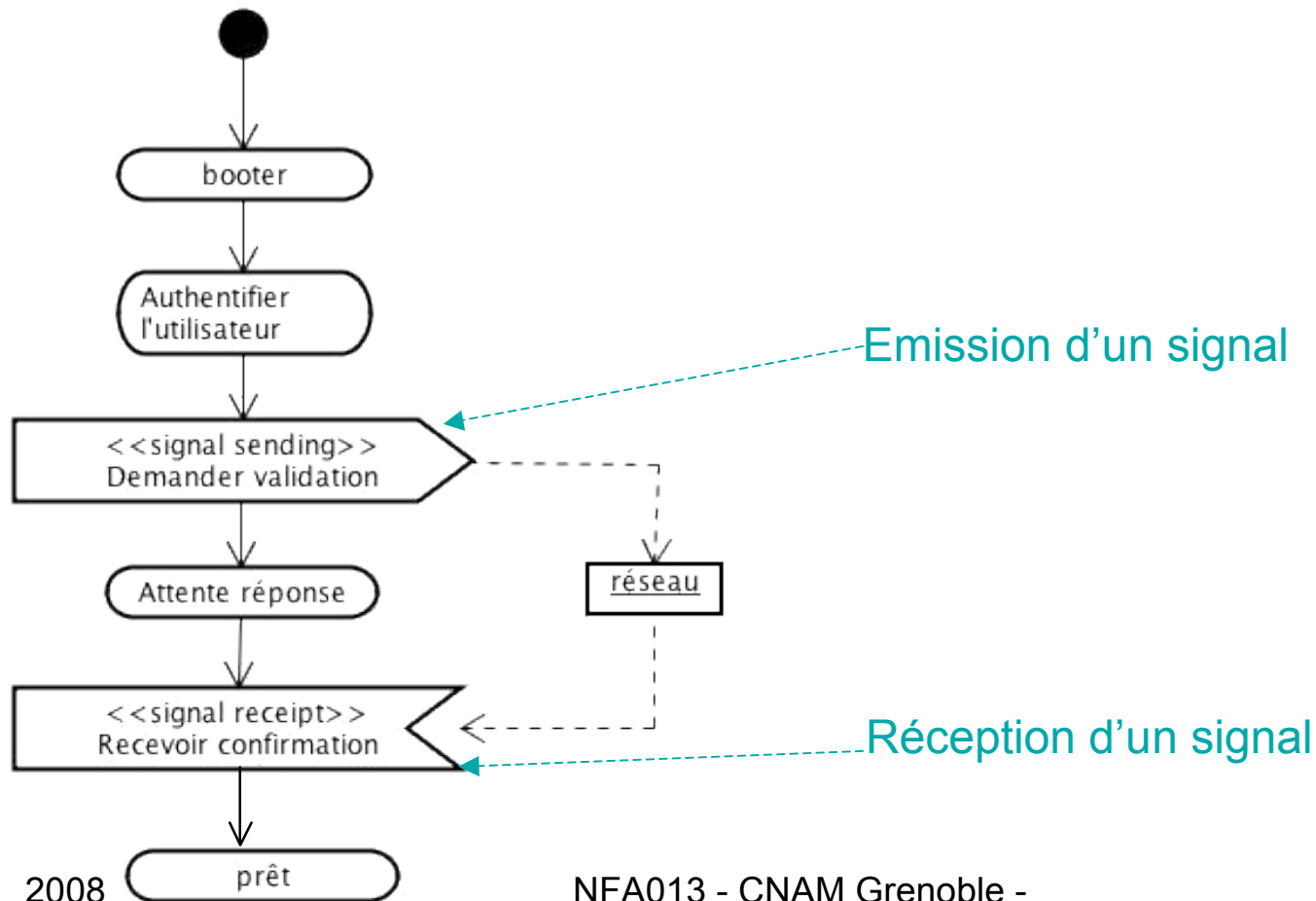


Diagramme d'activité

Emission et réception de signaux



Construire un diagramme d'activité

1. Identifier les activités

- Qu'arrive-t-il lorsqu'un médecin souhaite saisir une prescription ?
 - Créer dossier médical du patient
 - Créer un dossier pour les antécédents
 - Saisir la prescription

2. Ordonner les activités en utilisant les transitions

Attention, le niveau de détails des activités d'un même diagramme doit être identique.

Diagramme d'activité : exemple

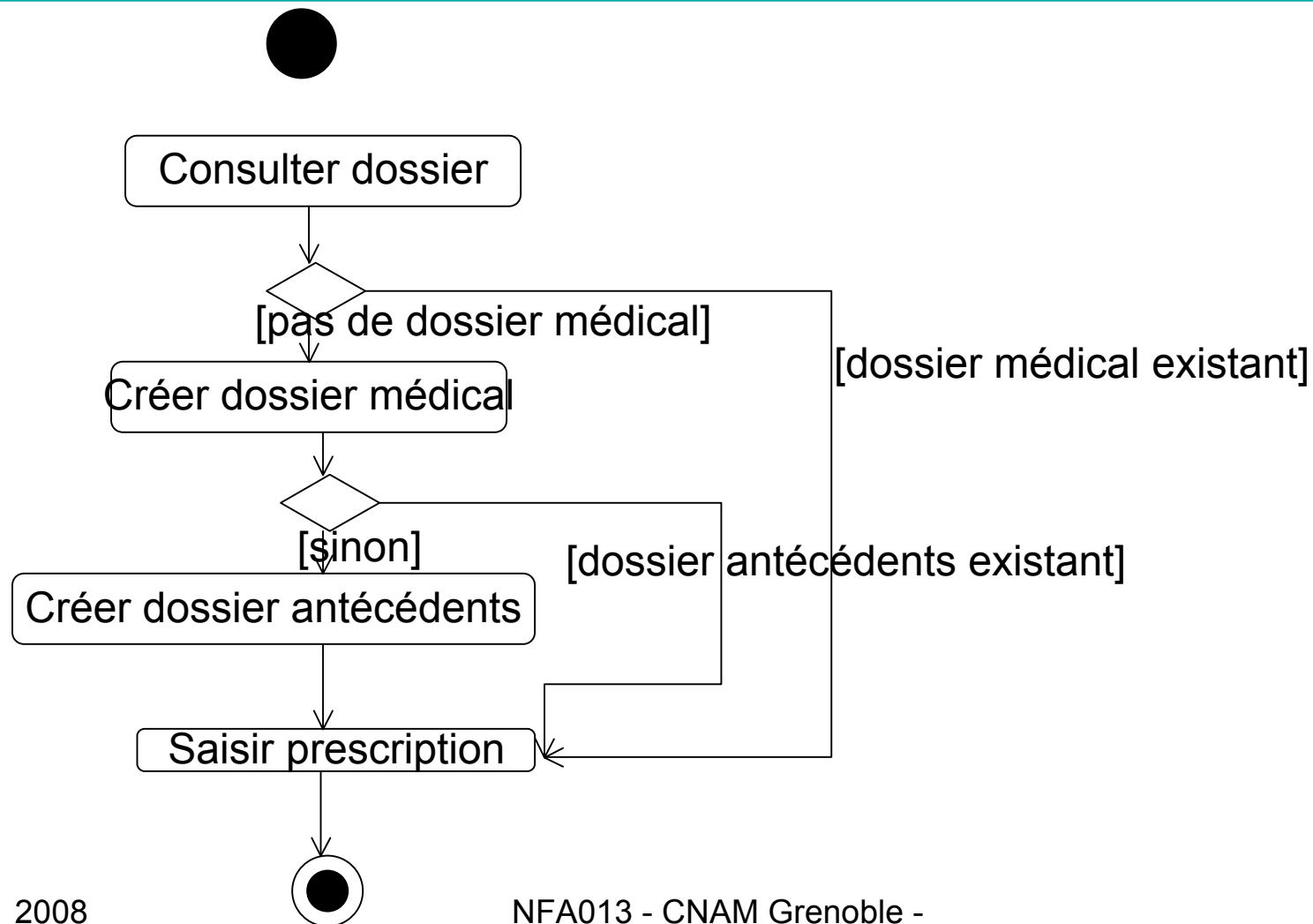


Diagramme d'activité : conseils

1. L'objectif ? Décrire un workflow, des opérations, un algorithme, un flux d'objets.
2. Les critères de qualité ?
 - Un niveau de détails équilibré dans chaque branche
 - Les branchements sont tous documentés et couvrent le domaine. Usage du [sinon]
3. Le niveau de détail souhaité ?
 - Haut niveau d'abstraction ou plutôt détaillé ?

Bilan diagramme d'activité

Nous avons vu :

- L'utilité d'un diagramme d'activité
- Quelles sont les notations associées à un diagramme d'activité
- Comment construire un diagramme d'activité

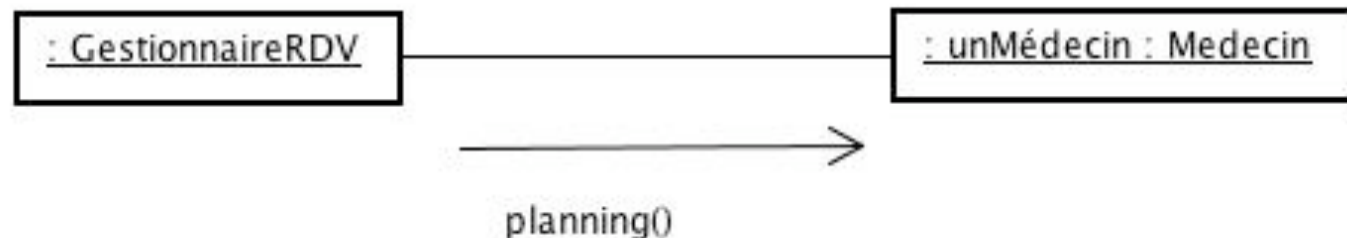
Les interactions entre objets

1. L'envoi de message

- Les objets communiquent par envoi de messages

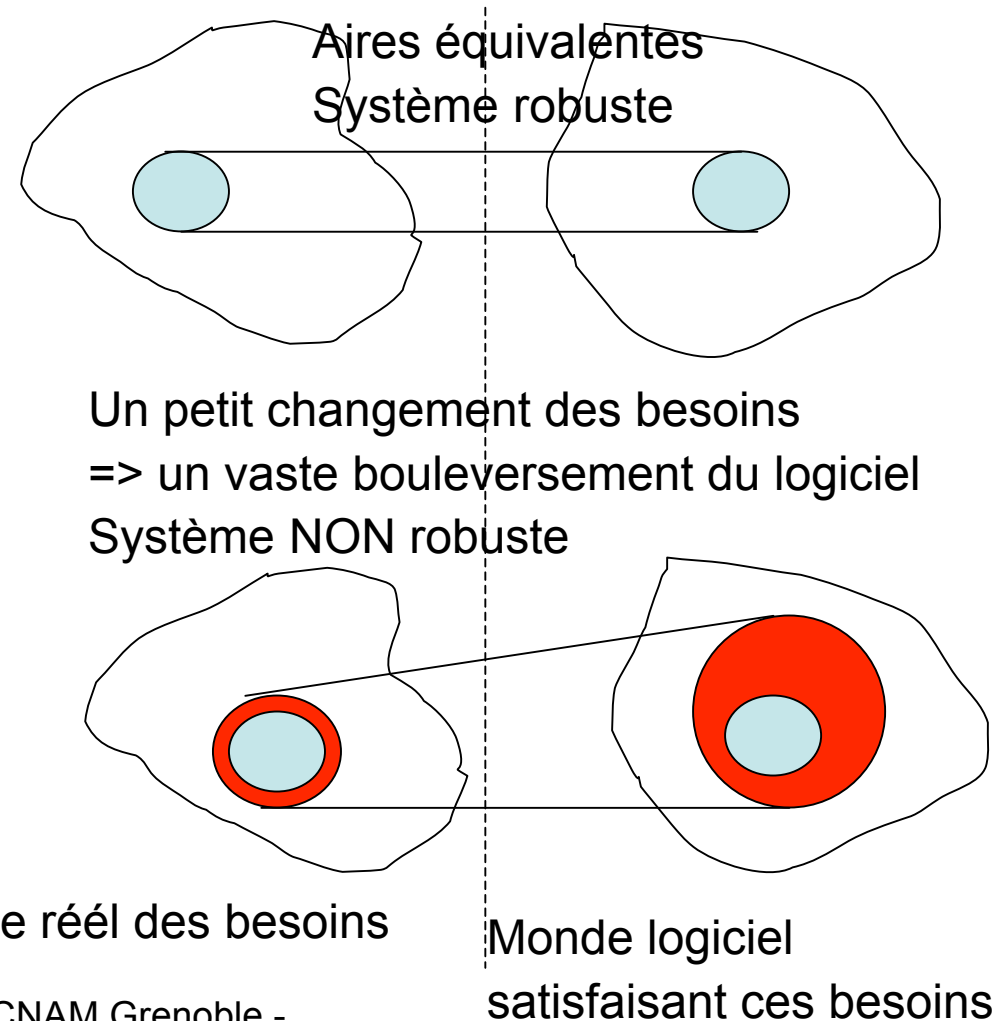
Envoyer le message `fournir_planning()` à un objet `UnMédecin`, doit utiliser la syntaxe suivante :

```
planning = UnMédecin.planning()
```



L'envoi de messages

- Les objets communiquent à travers l'envoi de messages
- L'objectif des diagrammes d'interaction est de **préciser les responsabilités** des objets pour minimiser l'effet de bord résultant des changements des besoins.



Interaction et collaboration

- On appelle *collaboration* l'ensemble formé par:
 - la structure des instances participant à un comportement
 - les liens qu'elles entretiennent.
- Une *interaction* est définie dans le contexte d'une collaboration.

Elle spécifie le canevas de communication entre les rôles dans la collaboration. Elle contient un ensemble de *messages* partiellement ordonné. Chaque message spécifie une communication comme par exemple le signal à envoyer ou l'opération à invoquer, ainsi que les rôles joués par la source et la cible du message.

Diagramme de séquence

- Les interactions entre objets selon un point de vue *temporel*
- Le contexte des objets n'est pas représenté de façon explicite. La représentation se concentre sur l'expression des interactions
- Peuvent être utilisés à différentes phases du cycle de vie avec +/- de détails
- Peuvent être utilisés de façon :
 - Informelle : documentation des cas d'utilisation en *illustrant les scénarios*
 - Informatique : le concept de MESSAGE unifie toutes les formes de communication entre objets (appel de procédure, événement discret, signal entre flots d'exécution)

Diagramme de séquence

- La dimension verticale représente le *temps*
- Les *objets* impliqués sont disposés horizontalement. Ils sont représentés par des *lignes de vie* verticales
- Les *messages* sont représentés par des flèches pleines horizontales
- L'exécution d'une opération est matérialisée par une *activation*

Diagramme de séquence : notation

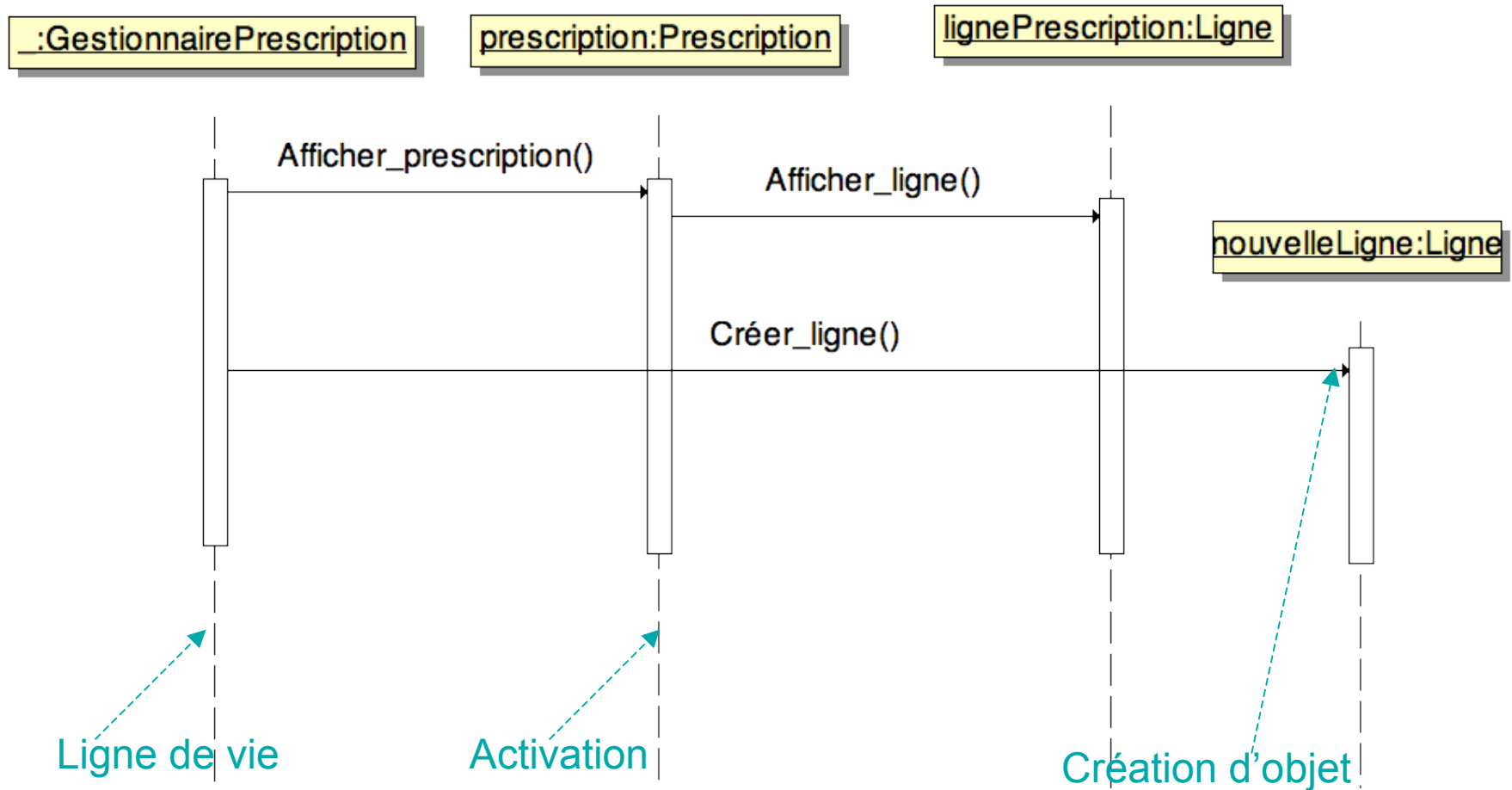


Diagramme de séquence : création/destruction d'objets

1. Création : pointer le message de création sur le rectangle symbolisant l'objet créé.
2. Destruction : fin de la ligne de vie, et un croix X.

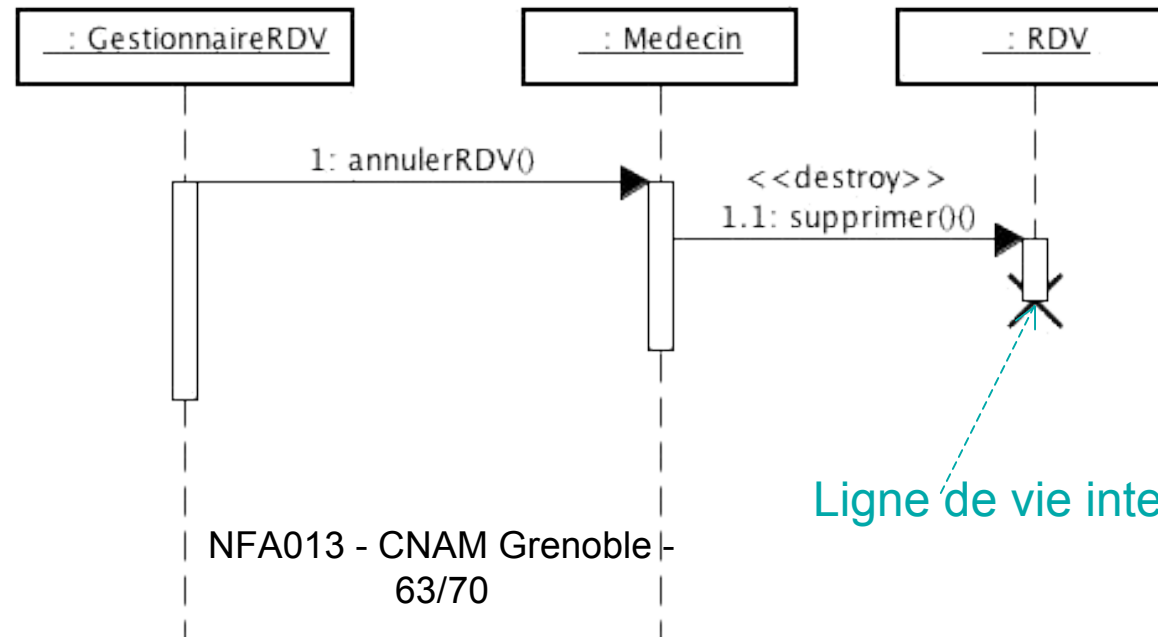


Diagramme de séquence : les différents types de message




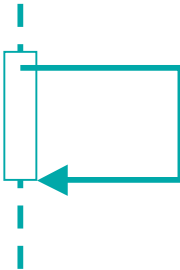
1. Message synchrone : l'appelant est bloqué en attente du retour de message 
2. Message asynchrone : l'appelant continue son flot d'exécution après l'émission. La réception se fait ultérieurement. 
3. Retour de procédure : le retour des messages sont le plus souvent implicite, mais parfois ils sont indiqués (cas asynchrone) 
4. Message réflexif : les objets peuvent produire des messages pour eux-mêmes. 

Diagramme de séquence : les contraintes temporelles

- Instant d'émission x
- Instant de réception x'
- Exemples :
 - Il s'écoule moins d'une seconde entre l'émission de $m1$ et $m2$
 - Il s'écoule au moins 2 secondes entre l'envoi et la réception de $m3$

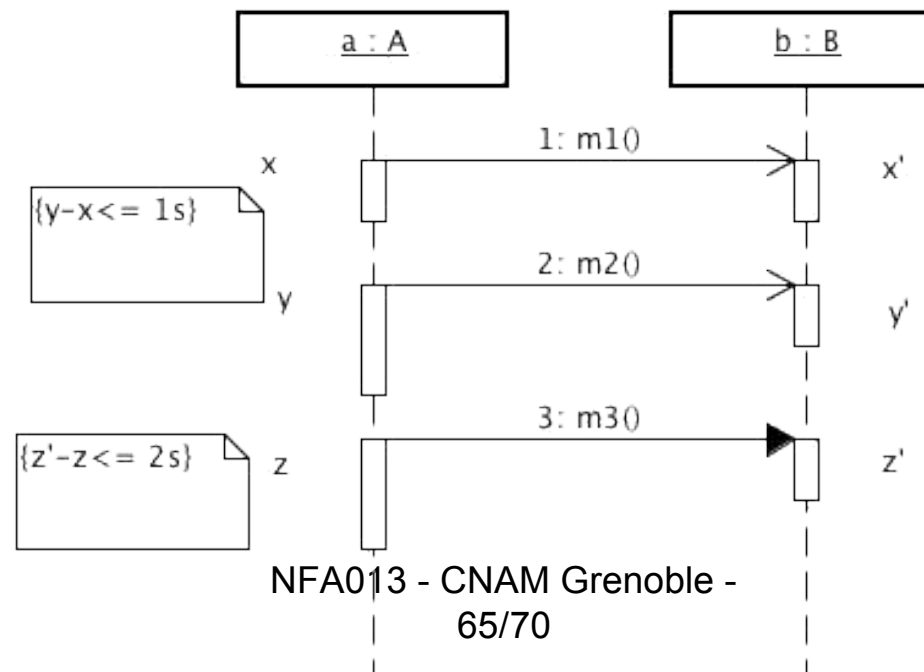


Diagramme de séquence

Exemples de messages synchrones/asynchrones

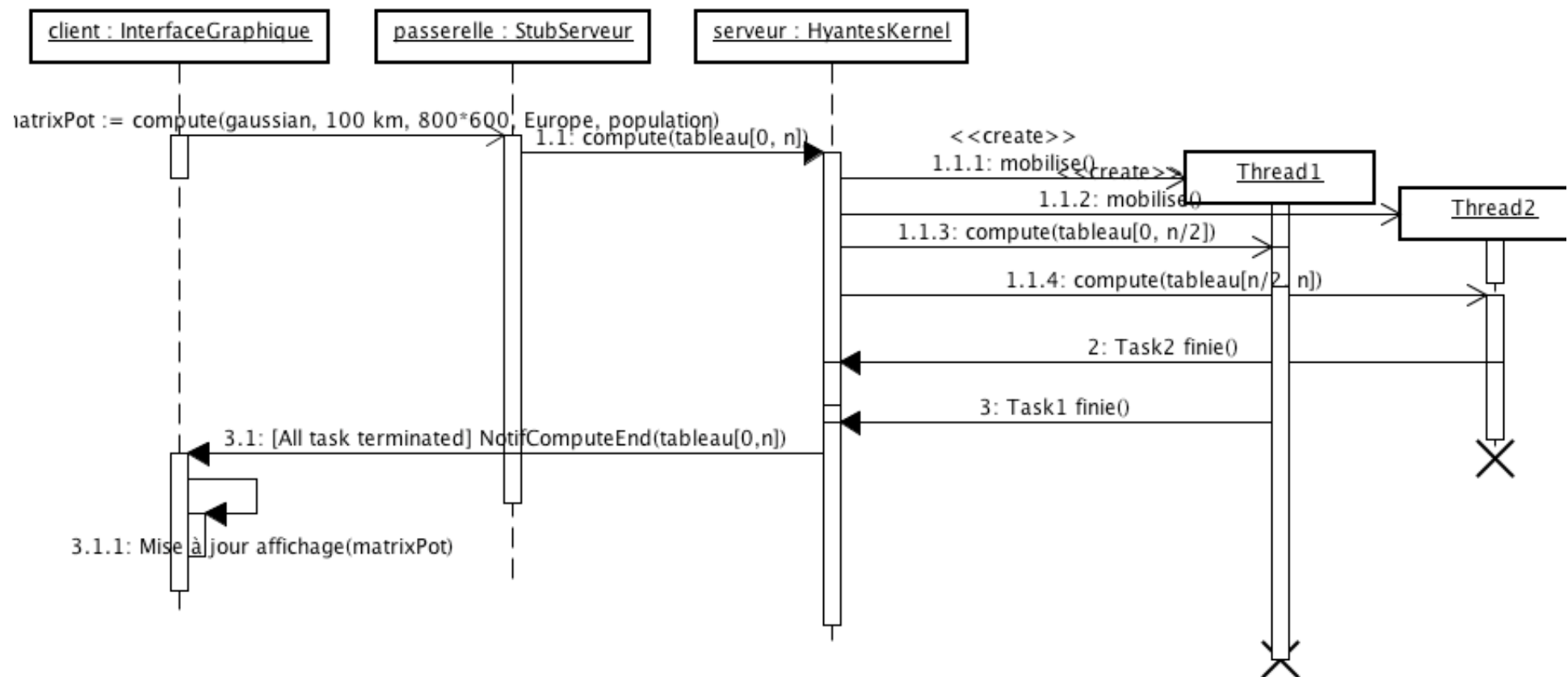


Diagramme de séquence : les branchements conditionnels

A émet un message

soit vers B si la condition 1 est remplie,

soit vers C si la condition 2 est remplie

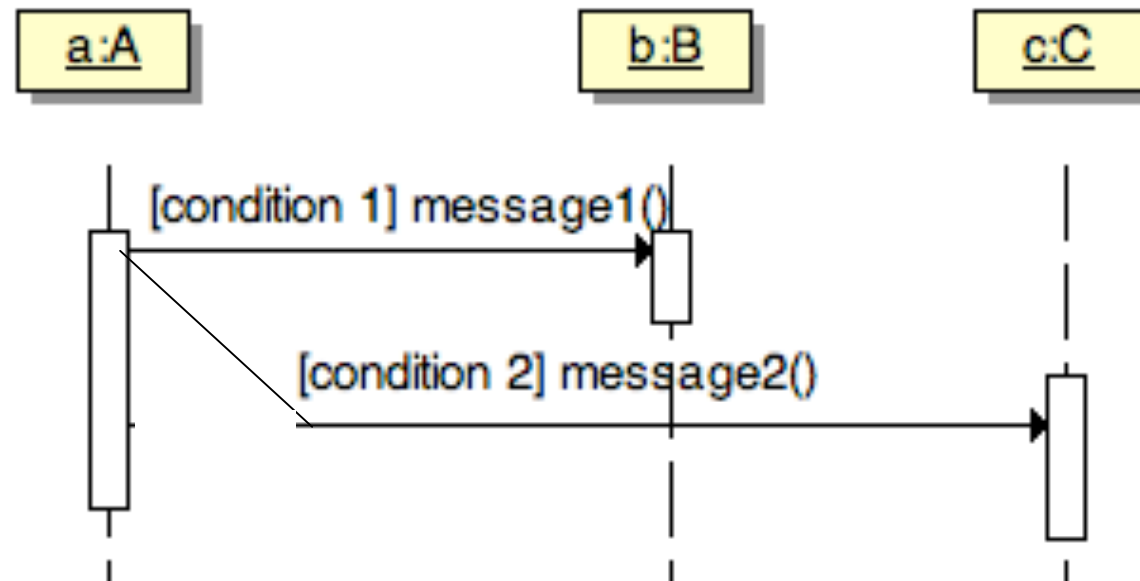
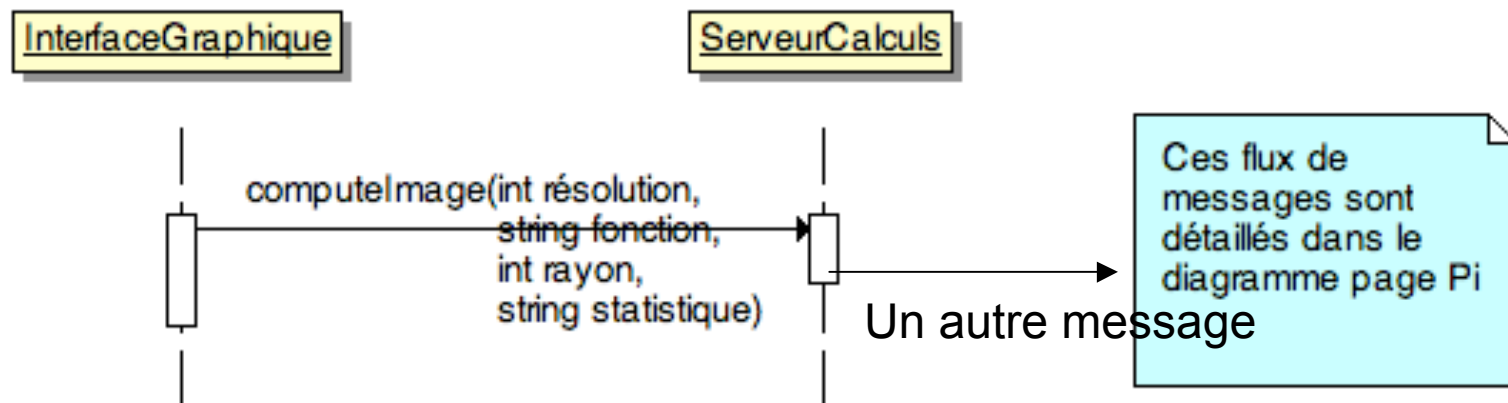


Diagramme de séquence : gérer la complexité

Les diagrammes complexes peuvent être décomposés en plusieurs diagrammes plus simples en précisant convenablement cette décomposition par des annotations.



Les diagrammes de séquence : conseils pour la description des besoins

1. Un scénario par cas d'utilisation : évitez les branchements conditionnels
2. Subdiviser les interactions complexes : décrivez dans un second diagramme une sous-tâche.
3. Un diagramme de séquence par condition d'erreur possible.

Les diagrammes de séquence : exemple

Service téléphonique : interaction avec l'opérateur

