Université de Jijel
Faculté des sciences exactes et d'informatique
Département d'informatique
Classe: 2ème Master SIAD

Cours en Système d'information, méthodes avancées

Chapitre 5: Analyse

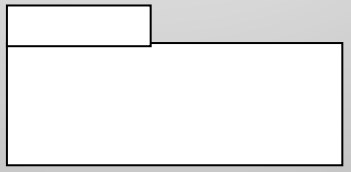
Présenté par: Dr. D. Boukraa
Maitre de conférences en informatique
boukraa.jimdofree.com
Année 2020-2021

Objectif du chapitre

- Savoir approfondir l'étude des besoins fonctionnels et techniques identifiés dans les étapes précédentes
- Savoir regrouper les classes en catégories pour faciliter leur développement
- Savoir corriger et affiner le modèle statique (diagramme de classes global)
- Savoir détailler le modèle dynamique (cas d'utilisation en le croisant avec le modèle statique)

1. Découpage en catégories du diagramme de classes

- Catégorie: application du concept de package UML pour regrouper des classes identifiées dans les diagrammes de classes candidates.
- Ensemble de classes fortement couplées (utilisée ensemble).
- Ce couplage résulte des dépendances entre les classes (associations, agrégation, composition, héritage ou échange de messages) ou d'autres facteurs.
- Une catégorie est représentée graphiquement comme un paquetage



1. Découpage en catégories du diagramme de classes

- Pourquoi découper?
  - meilleure représentation de la structure statique du système.
  - répartir les responsabilités des catégories sur plusieurs équipes de développement
  - réutiliser les catégories dans d'autres systèmes et de faciliter l'évolutivité, la réutilisation et la maintenance des composants du système.

1. Découpage en catégories du diagramme de classes

- Critères de découpage

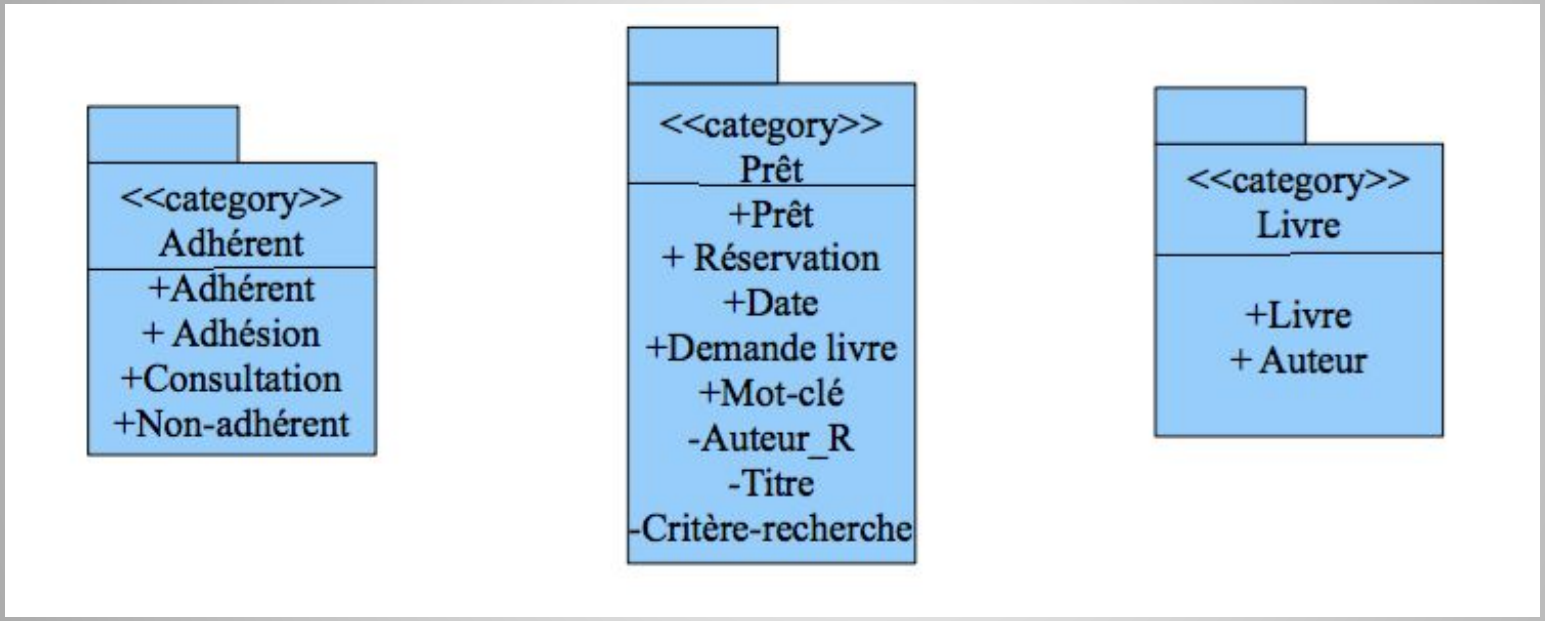
Le point de départ est l'ensemble des diagrammes de classes.

Les critères de découpage peuvent être

- *Cohérence des classes* : classes sont liées sémantiquement
- *Evolution* : en regroupant les classes a fort degré d'évolution ensemble et vice-versa
- *Durée de vie des objets* : on regroupe des classes dont les objets ont des durées de vies proches
- *Dépendance entre les classes* : la notion de dépendance est décrite plus loin

1. Découpage en catégories du diagramme de classes

- Exemple de catégories : *Bibliothèque*



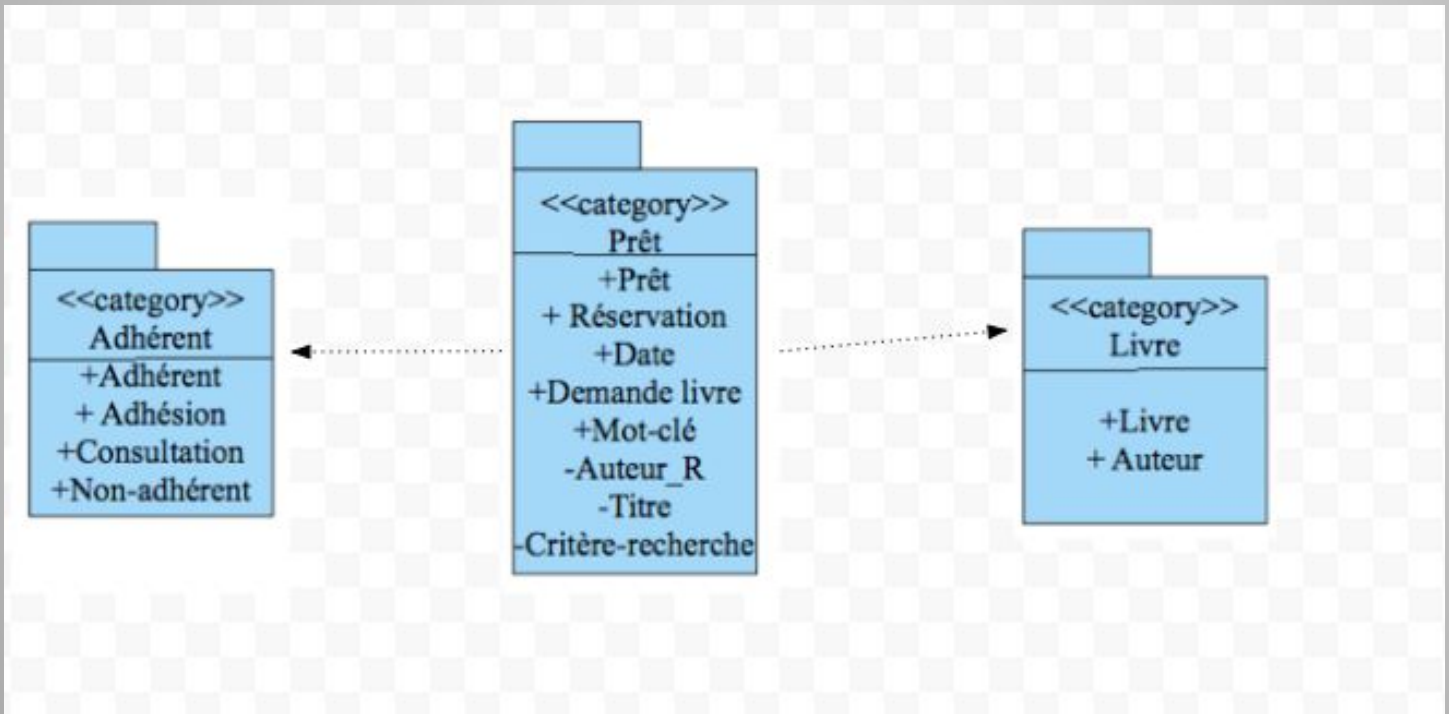
1. Découpage en catégories du diagramme de classes

- Dépendance entre les catégories
  - Liée à la notion d'utilisation de classes en dehors de leurs catégories d'origine.
  - Lorsqu'une association existe entre deux classes A et B de catégories différentes, si à partir de la classe A on veut accéder à la classe B, cela se traduit par une dépendance entre les catégories: *A dépend de B*.
  - Nécessité de visibilité de la classe à l'extérieur de sa catégorie. Dans ce cas, le nom de la classe est précédé par le symbole « + ».
  - L'utilisation de classe à partir dans les catégories autres que celle d'origine est symbolisée par le stéréotype <<import>>.

1. Découpage en catégories du diagramme de classes

- Dépendance entre les catégories

Exemple:



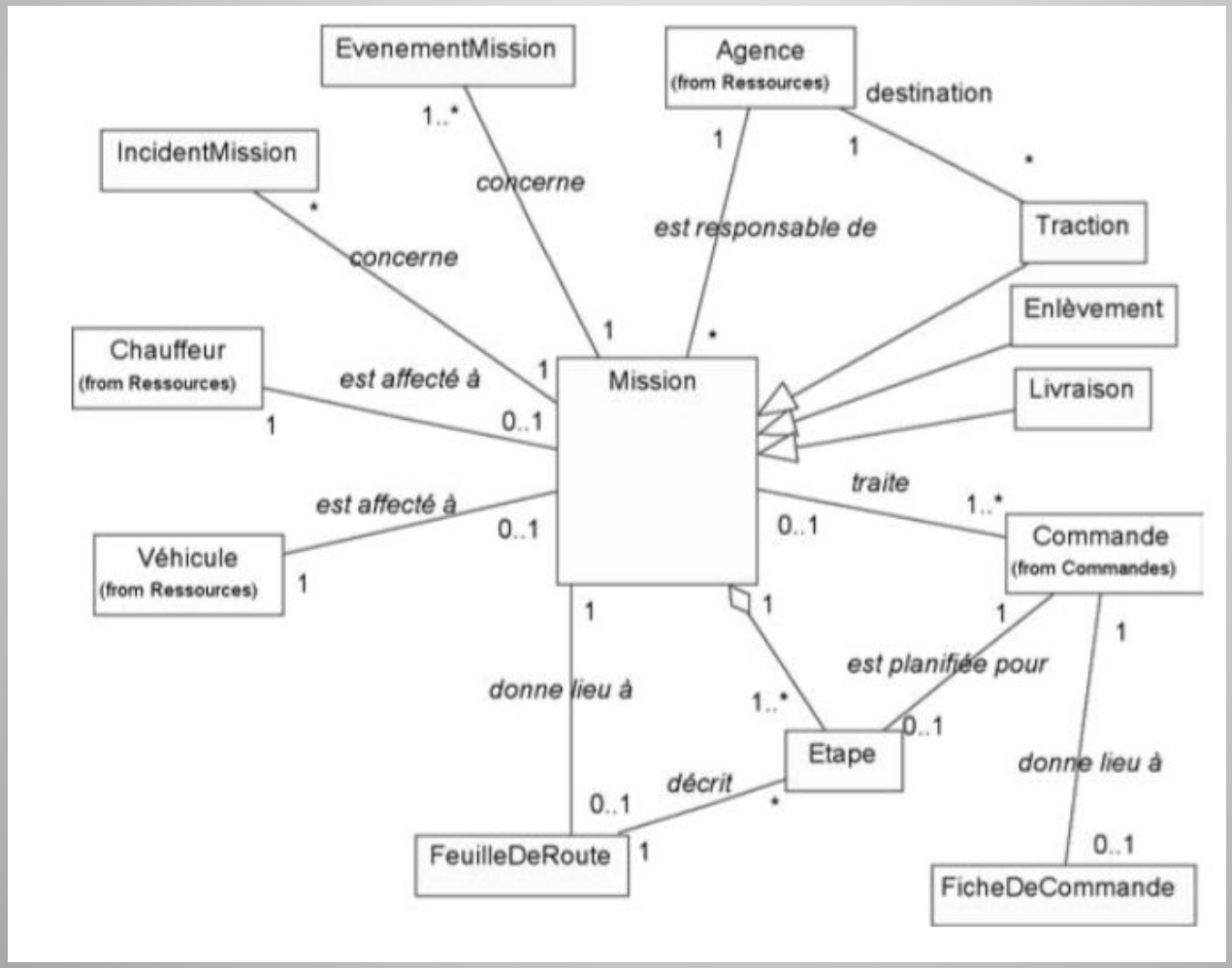
1. Découpage en catégories du diagramme de classes

- Diagramme de classes par catégories
  - Représenter les classes candidates d'origine de la catégorie
  - Ajouter les classes utilisée à partir des autres catégories à travers les associations
  - Utiliser le mécanisme d'importation (représenté par *{from ….}*) pour spécifier la catégorie d'origine de la classe.

1. Découpage en catégories du diagramme de classes

- Diagramme de classes par catégories

Exemple (référence: UML 2 en action, P. Roques, Eyrolles 4ème Ed. 2007



1. Découpage en catégories du diagramme de classes

- Résultat du découpage en catégories
  - La liste des catégories
  - Le diagramme de packages
  - Le diagramme de classes par catégorie

2. Développement du modèle statique

- Le découpage en catégorie génère un regroupement et répartition des classes candidates.
- Ces classes doivent être affinées et les anomalies traitées
- Aspects de traitement:
  - Classes et associations
  - Attributs des classes
  - Opérations des classes

2. Développement du modèle statique

- Anomalies au niveau classes et associations:
  - *Classes redondantes:* plusieurs classes représentant le même concept. Ex: employé et personne.
  - *Classes à la place d'attributs*. Ex: numéro de téléphone
  - *Classes à la place de rôles.* Classes traduisant des associations entre classes. Ex: voiture_conduite.
  - *Plusieurs classes ayant le même nom* mais représentant des concepts différents. Ex: personne.

2. Développement du modèle statique

- Anomalies au niveau classes et associations:
  - *Classe représentant des acteurs*. Ex: classe agent de saisie.
  - *Classe techniques* (de conception): classes représentant des objets techniques. Ex: fichier
  - *Casse de groupes ou listes d'objets*: utilisés pour représenter les grilles. Ex: liste des étudiants
  - *Classes grosses*: classe représentant plusieurs concepts de la réalité ou traduisant le contenu des documents.

2. Développement du modèle statique

- Aspects à vérifier au niveau classes et associations:
  - *Multiplicité*
  - *Navigabilité*
  - *Sens des associations (agrégation, compositions)*
  - *Ajout de contraintes*
  - *Ajout de stéréotypes*

2. Développement du modèle statique

- Anomalies et aspects à vérifier au niveau attributs
  - *Multiplicité des attributs*. Un attribut peut avoir plusieurs valeurs pour le même objet de classe.
  - *Attributs pouvant représenter des classes*. Les regrouper avec d'autres attributs pour former de nouvelles classes.
  - *Attributs dérivés*. Les précéder du symbole « / ».
  - *Ajout de stéréotypes et de contraintes*. Ex: clés primaires, intervalles de valeurs, etc.

2. Développement du modèle statique

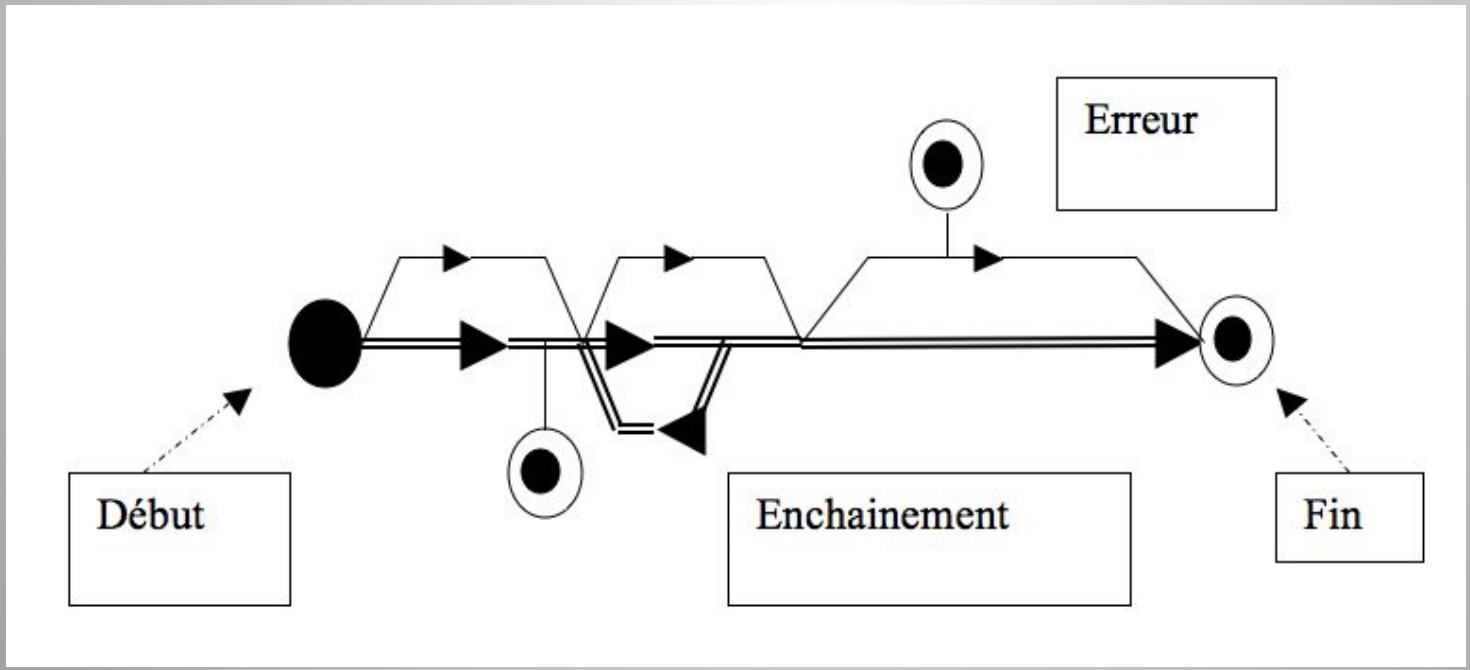
- Aspects à vérifier au niveau des opérations
  - S'intéresser à cette étape aux opérations de gestion (se référer aux cas d'utilisation)
  - Écarter les opérations de base:
    - Création et destruction d'instances
    - Lecture et mise à jour des valeurs d'attributs
    - Création et destruction des liens entre les instances d'associations
    - Recherche des instances d'associations
    - Opérations IHM (éditer, imprimer, etc.)

3. Développement du modèle dynamique

- Se fait en parallèle que le développement du modèle statique.
- Concerne le détail de la description des cas d'utilisation par d'autres diagrammes.
- A cette étape
  - On identifie les scénarios
  - On formalise les scénarios
  - On formalise les changements d'état des objets de classes
  - On complète le diagramme de classes par les attributs et les opérations identifiées à l'analyse dynamique

3. Développement du modèle dynamique

- Identification des scénarios:
  - Scénario: exécution particulière d'un cas d'utilisation
  - Un cas d'utilisation comporte au moins un scénario



3. Développement du modèle dynamique

- Identification des scénarios:
  - Lorsqu'un cas d'utilisation comporte plusieurs combinaisons d'enchaînements, choisir un minimum de combinaisons couvrant ensemble tous les enchaînements.
  - Se servir des diagrammes d'activité pour trouver un ensemble minimal de scénarios.

3. Développement du modèle dynamique

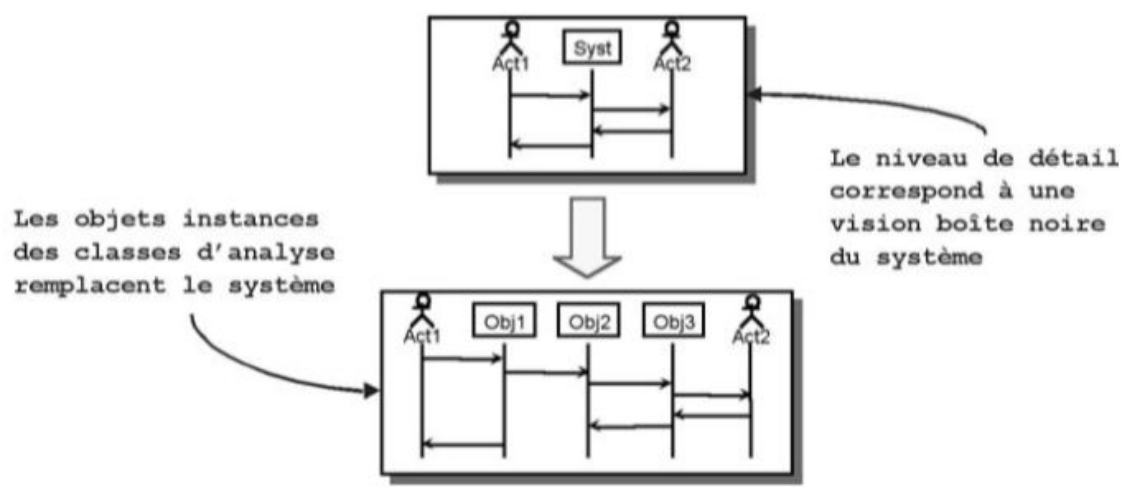
- Identification des scénarios:
  - On distingue quatre types de scénarios
    - *Nominaux :* scénarios qui permettent d'atteindre la fin du cas (et en conséquent réaliser la post-condition) de manière naturelle.
    - *Alternatifs :* enchainement qui aboutissent à la fin du cas mais dans des situations rares car il s'agit d'alternatives en face à des exceptions du traitement normal.
    - *Aux limites :* définissent les limites d'exécution du cas de telle sorte qu'une autre exécution du cas provoquerait une erreur.
    - *D'exception :* sorties anormales du cas d'utilisation qui n'aboutit pas aux post-conditions

3. Développement du modèle dynamique

- Identification des scénarios:
  - Exemples de types de scénarios (bibliothèque)
    - *Scénarios nominaux*
      - Création de l'adhérent d'une adhésion
      - Ajout d'un prêt, d'une restitution
    - *Scénarios alternatifs*
      - Modification des informations sur un adhérent
    - *Scénario à la limite*
      - Ajout de prêt du dernier livre (atteinte de la limite) à un adhérent
    - Scénario d'exception
      - Ajout d'un prêt après avoir atteint la limite

3. Développement du modèle dynamique

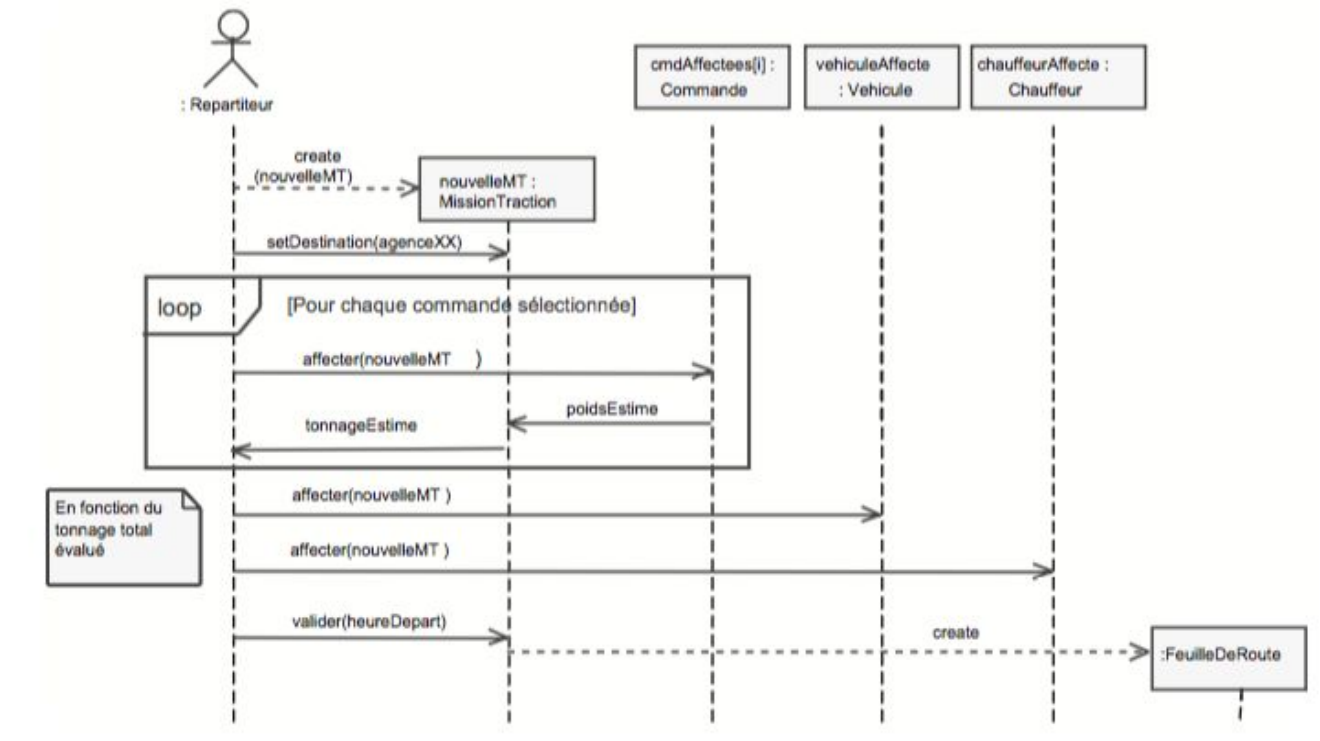
- Formalisation des scénarios:
  - Pour chacun des scénarios sélectionner, le formaliser par
    - Un diagramme de séquences : remplacer le système (boite noire) par l'ensemble des objets impliqués (réf fig: UML 2 en action)



    - Un diagramme de communication

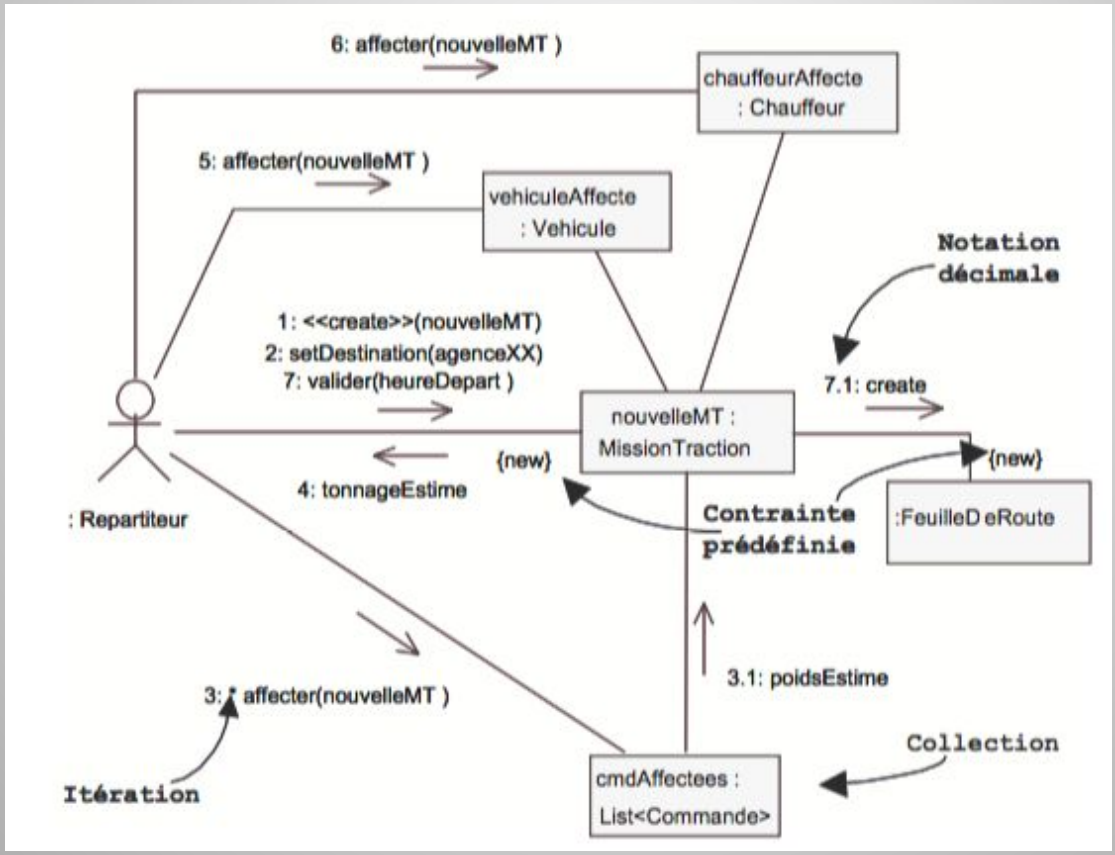
3. Développement du modèle dynamique

- Formalisation des scénarios:
  - Exemple de formalisation par un diagramme de séquences (réf: UML 2 en action)



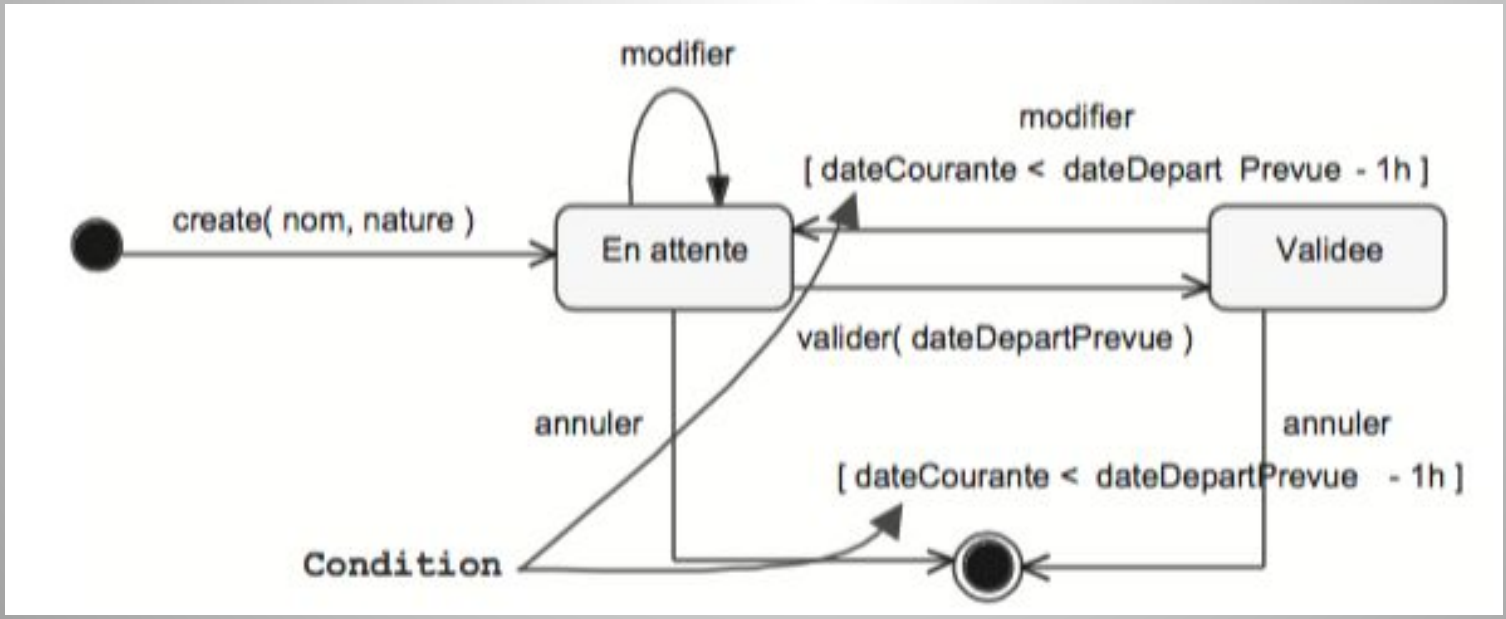
3. Développement du modèle dynamique

- Formalisation des scénarios:
  - Exemple de formalisation par un diagramme de communication (réf: UML 2 en action)



3. Développement du modèle dynamique

- Formalisation des changements d'état des objets:
  - Permet de représenter les changements d'état des objets suite à des évènements et en respectant des conditions.
  - Exemple de diagramme d'état-transition (réf: UML 2 en action)



3. Développement du modèle dynamique

- Compléter le diagramme de classes par les attributs et opérations
  - Se servir des
    - Diagrammes de séquences
    - Diagrammes de communication
    - Diagramme d'états-transitions