

Un processus pour la conception orientée objet

Frédéric Migeon

Prerequis

- Programmation Orientée Objet
 - Objet - Classe - Héritage - Délégation
 - Objet - Message - Méthode - Communication synchrone
- UML
 - Diagrammes de classes
 - Diagrammes d'activités
 - Diagrammes de séquences
 - Diagrammes d'états-transitions

Introduction - Vocabulaire

- Ingénierie des méthodes
 - Méthode
 - Processus
 - Modèles
 - Outils
- Vocabulaire
 - Processus : ensemble ordonné de phases
 - Phase : ensemble ordonné d'activités
 - Activité : ensemble ordonnée d'étapes
 - Etape : élément atomique combinant des ressources/artefacts pour en produire de nouveaux
- Hypothèse de base sur le "client"
 - Incomplet
 - Ambigu

Introduction - Où commence-t-on ?

- Les protagonistes
 - "Client" (aka les parties prenantes)
 - "Le Fournisseur" (aka l'équipe de dev)
- L'appel d'offre
 - Le cahier des charges
 - La réponse
- Le kick-off
 - Véritable démarrage
 - Origine du processus "MCO"

Introduction - Mise en garde

- Un processus en cascade ?
 - Etapes et activités
 - >
 - Phases linéaires et séquentielles
- Un processus en 3 phases ?
 - Focus sur la conception
 - Sans dénigrer l'avant et l'après
- Dessiner des diagrammes ?
 - Processus, focus sur les activités
 - UML n'est qu'un support langagier
- UML et Design Patterns ?
 - Of course but... time is money :-/
 - Once upon a time : 150h -> 50h de conception
- Guide en TP et professionnel ?
 - La pratique de référence
 - Les principes de la pratique
 - L'adaptation de la pratique

Plan

1. Introduction - Vocabulaire
2. Ph1-Modélisation du métier
 - 2.1. A1-Modéliser les processus métiers
 - 2.2. A2-Modéliser le domaine métier
3. Ph2-Modélisation des besoins
 - 3.1. A3-Identifier les limites
 - 3.2. A4-Identifier les fonctionnalités
 - 3.3. A5-Modéliser le périmètre fonctionnel
 - 3.4. A6-Décrire le périmètre fonctionnel de façon détaillée
4. Ph3-Réalisation des besoins
 - 4.1. A7-Réaliser chacune des fonctionnalités
 - 4.2. A8-Factoriser & Réfactorer
 - 4.3. A9-Planter

Phase 1 - Modélisation métier

A1-Modéliser les processus métiers

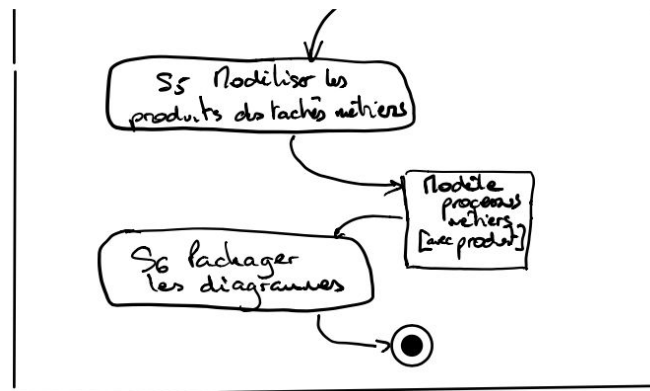
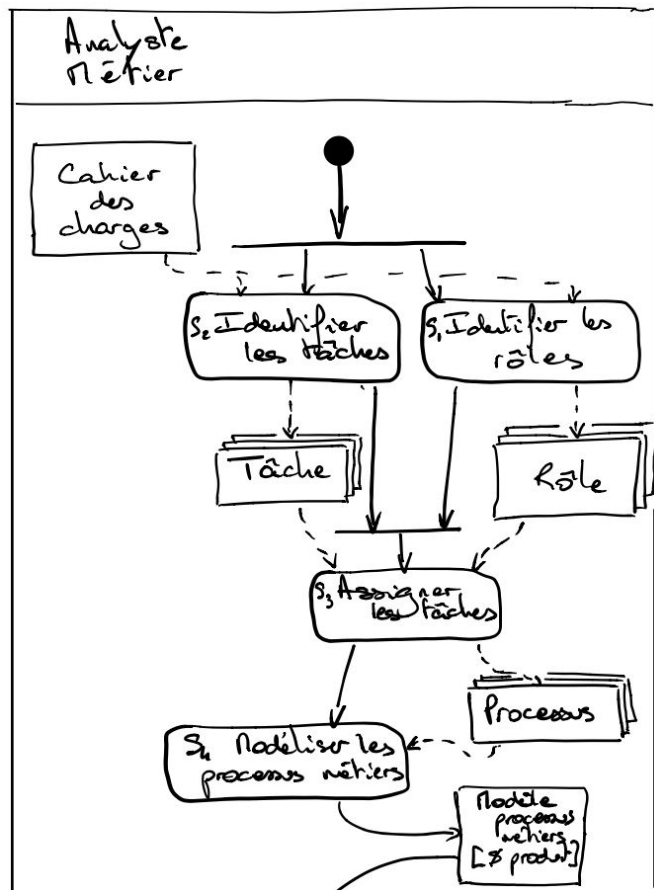
A2-Modéliser le domaine métier

A1 - Modéliser les processus métiers

1. A1-S1 Identifier les rôles/responsabilités
 2. A1-S2 Identifier les tâches
 3. A1-S3 Assigner les tâches
 4. A1-S4 Modéliser les processus métiers
 5. A1-S5 Modéliser les produits des tâches métiers
 6. A1-S6 Packager les diagrammes
- Conseils et pratique
 - Pas de système logiciel
 - Modéliser l'existant
 - Être complet pour être détaillé pour être cohérent et homogène
 - Conseil de modélisation (UML)
 - Diag. Activités (DA)
 - Swimlane \Leftrightarrow Rôle
 - Activité \Leftrightarrow tâche métier
 - Objet \Leftrightarrow produit métier
 - Règles de forme valide
 - [Moodle](#)

Consultez les règles de bonne modélisation sur Moodle !

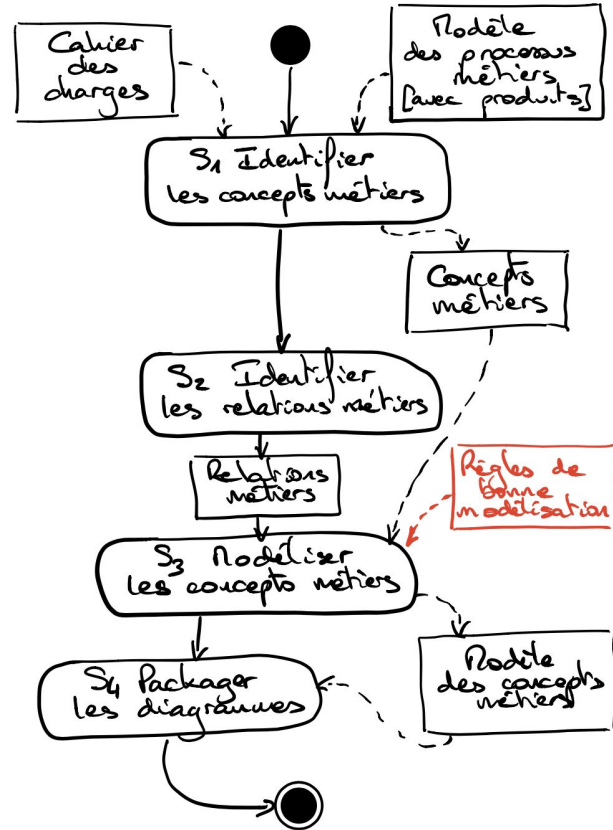
Résumé graphique



A2 - Modéliser le domaine métier

1. A2-S1 Identifier les concepts métiers
 2. A2-S2 Identifier les relations métiers
 3. A2-S3 Modéliser les concepts métiers
 4. A1-S4 Packager les diagrammes
- Conseils et pratique
 - A partir des produits métiers
 - Tous les produits
 - Pas du code !
 - Abusez des rôles (UML)
 - Conseil de modélisation (UML)
 - Diag. de Classes (DC Domaine)
 - Classe \Leftrightarrow Produit
 - Association/Agrégation/Composition
 - Classe d'association
 - Généralisation/Spécialisation
 - Règles de forme valide
 - [Moodle](#)

Résumé graphique



Phase 2 - Modélisation des besoins

A3-Identifier les limites

A4-Identifier les fonctionnalités

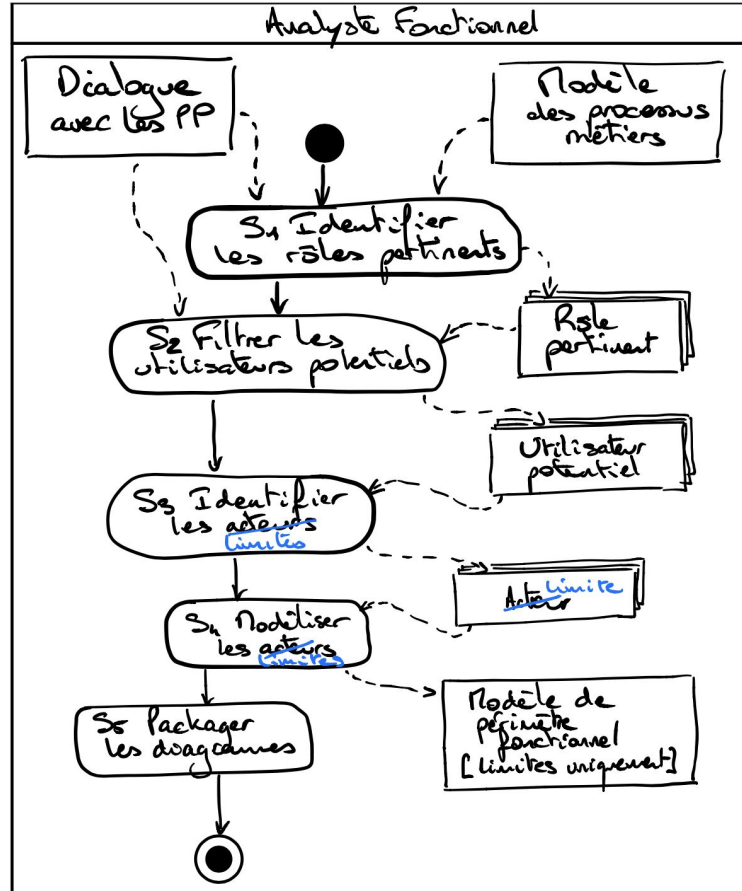
A5-Modéliser le périmètre fonctionnel

A6-Décrire le périmètre fonctionnel de façon détaillée

A3 - Identifier les limites

1. A3-S1 Identifier les rôles pertinents
 2. A3-S2 Filtrer les utilisateurs potentiels
 3. A3-S3 Identifier les acteurs
 4. A3-S4 Modéliser les acteurs
 5. A3-S5 Packager les diagrammes
- Conseils et pratique
 - Utiliser l'expertise métier
 - Modèle des processus métiers
 - Dialogue avec les PP
 - Conseil de modélisation (UML)
 - Diag. Cas d'Utilisation (DCU)
 - Acteur primaire <=> Utilisateur
 - Généralisation entre acteurs
 - Règles de forme valide
 - [Moodle](#)

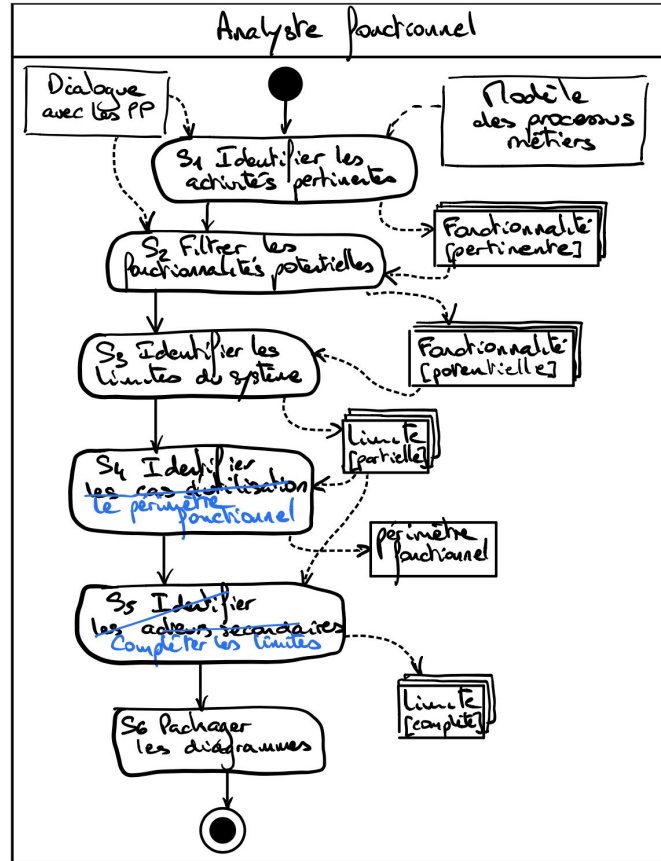
Résumé graphique



A4 - Identifier les fonctionnalités

1. A4-S1 Identifier les activités pertinentes
 2. A4-S2 Filtrer les fonctionnalités potentielles
 3. A4-S3 Identifier les limites du système
 4. A4-S4 Identifier les cas d'utilisation
 5. A4-S5 Identifier les acteurs secondaires
 6. A4-S6 Packager les diagrammes
- Conseils et pratique
 - Utiliser l'expertise métier
 - Modèle des processus métiers
 - Dialogue avec les PP
 - Conseil de modélisation (UML)
 - Diag. Cas d'Utilisation (DCU)
 - Cas d'utilisation \Leftrightarrow Fonctionnalité
 - Acteur secondaire \Leftrightarrow Extérieur
 - Généralisation/Extension/Inclusion
 - Règles de forme valide
 - [Moodle](#)

Résumé graphique



A5 - Modéliser le périmètre fonctionnel

1. A5-S1 Modéliser les limites
 2. A5-S2 Modéliser les fonctionnalités
 3. A5-S3 Généraliser
 4. A5-S4 Factoriser
 5. A5-S5 Packager les diagrammes
- Conseils et pratique
 - Utiliser l'expertise métier
 - Modèle des processus métiers
 - Dialogue avec les PP
 - Conseil de modélisation (UML)
 - Diag. Cas d'Utilisation (DCU)
 - Cas d'utilisation <=> Fonctionnalité
 - Acteur primaire <=> Utilisateur
 - Acteur secondaire <=> Extérieur
 - Généralisation/Extension/Inclusion
 - Règles de forme valide
 - [Moodle](#)

A6 - Décrire le périmètre fonctionnel de façon détaillée

Pour chaque fonctionnalité

Faire

1. A6-S1 Décrire textuellement avec détails
2. A6-S2 Modéliser le scénario nominal
3. A6-S3 Modéliser les alternatives
4. A6-S4 Modéliser les cas d'erreur

- Conseils et pratique
 - Utiliser l'expertise métier
 - Modèle des processus métiers
 - Modèle du périmètre fonctionnel
 - Dialogue avec les PP
- Conseil de modélisation (Texte)
 - Utilisation d'un template
 - Documenter pour indexer
 - Documenter pour préciser le besoin
- Conseil de modélisation (UML)
 - Diag. Séquence **Système**
 - Focaliser sur les interactions
 - Objet **Système** en approx. grossière
 - Acteurs primaires et secondaires
 - Messages <=> Interactions HM ou MM

A6S1 - Décrire textuellement avec un template

Titre

Description scenario nominal

But

Description

Acteur Primaire

[Acteurs secondaires]

Relations à d'autres CU

Pré-condition

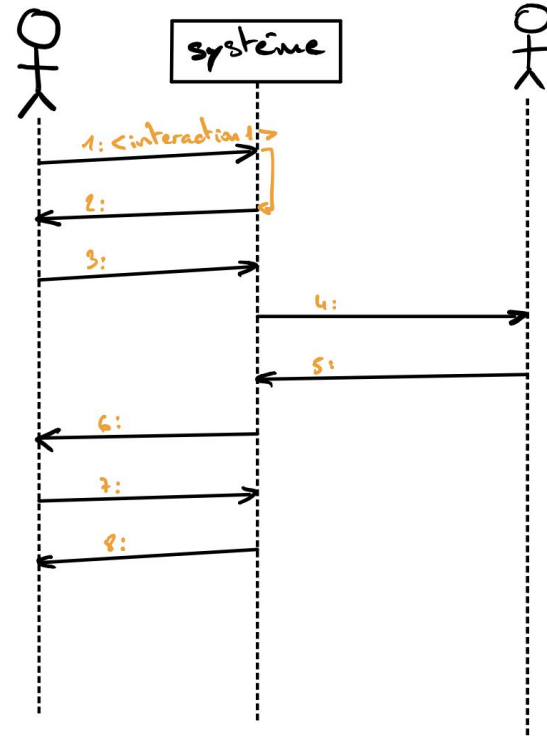
Post-condition

Exigences non fonctionnelles

Règles de gestion métier

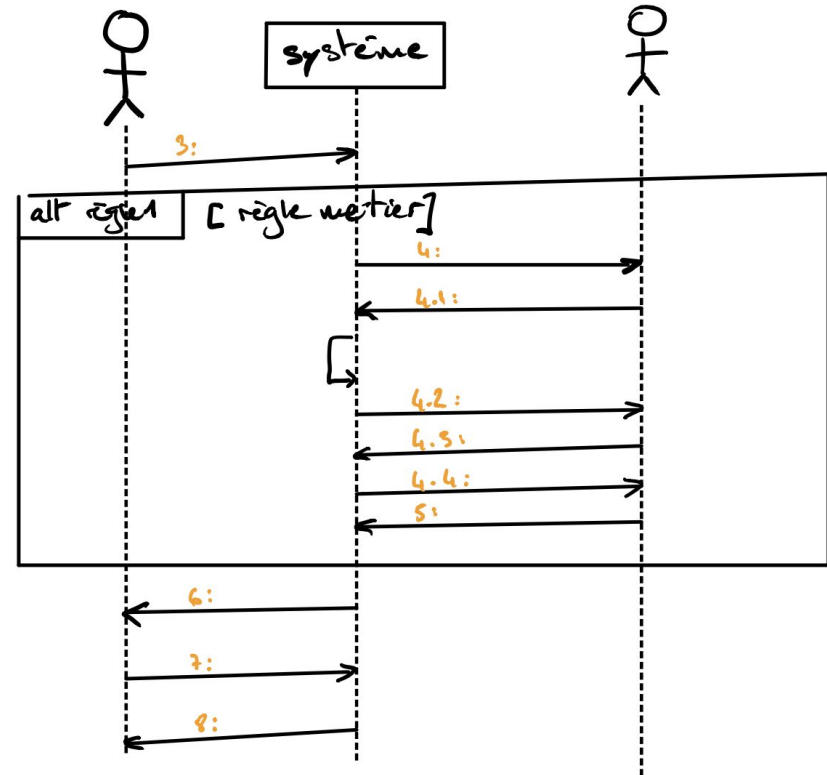
A6S2 - Modéliser le scénario nominal

- Conseils et pratique
 - première interaction depuis AP
 - des échanges (pas un monologue)
 - des labels
- Conseil de modélisation (UML)
 - Diag. séquence Système (DSS)
 - 1 seul objet système
 - les acteurs du DCU
 - [frame loop]
- Règles de forme valide
 - [Moodle](#)



A6S3/4 - Modéliser alternatives & cas d'erreur

- Conseils et pratique
 - partir d'un point d'alternative
 - identifier la règle métier
- Conseil de modélisation (UML)
 - Diag. séquence Système (DSS)
 - Frame opt et alt
- Règles de forme valide
 - [Moodle](#)



Phase 3 - Réalisation des besoins

A7-Réaliser chacune des fonctionnalités

A8-Factoriser & Réfactorer

A9-Planter

A7 - Réaliser chacune des fonctionnalités

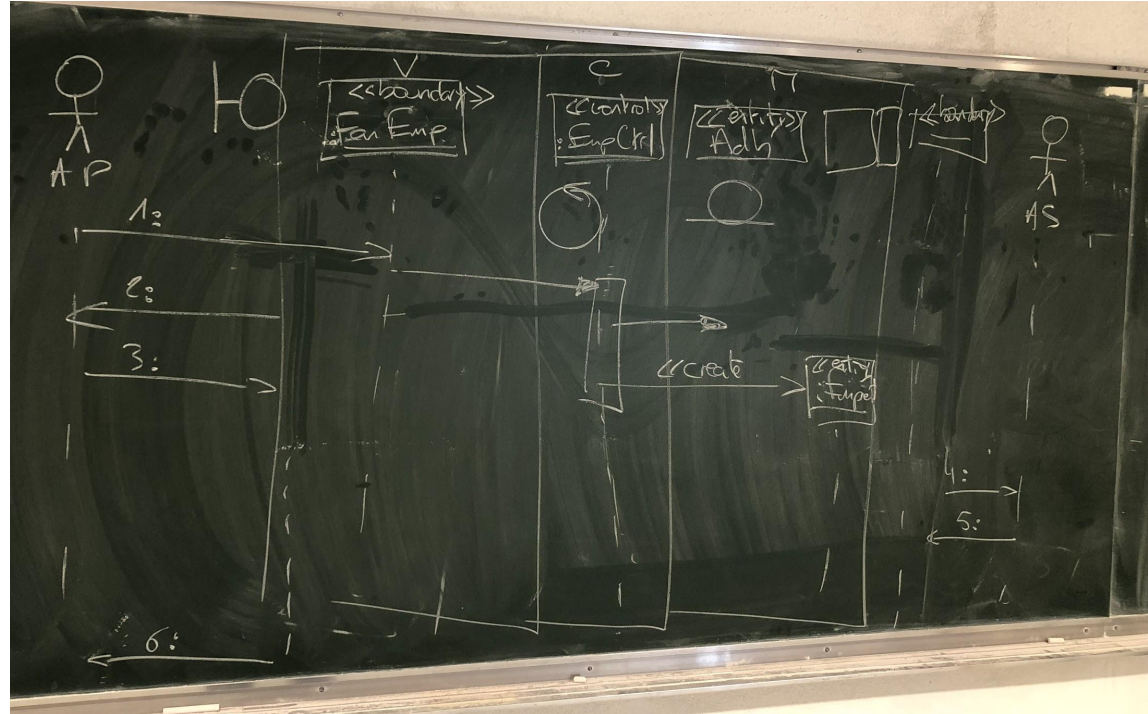
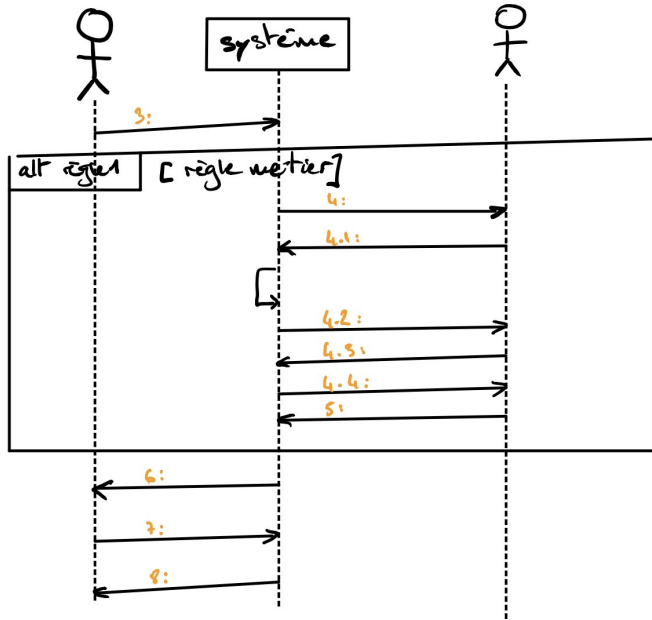
Pour chaque fonctionnalité

Faire

1. A7-S1 Récupérer les limites
2. A7-S2 Récupérer l'interfaçage
3. A7-S3 Définir l'architecture MVC initiale
 - a. Ajouter 1 objet <<boundary>> par acteur/limite
 - b. Ajouter 1 objet <<control>>
4. A7-S4 Rajouter les classes du domaine participantes
 - a. Identifier les classes
 - b. Taguer les objets <<entity>>
5. A7-S5 Concevoir en détails
 - a. Ajouter les objets "algorithmiques"
 - b. Définir les interactions entre objets
6. A7-S6 Modéliser la réalisation détaillée de la fn^{alité}
7. A7-S7 Compléter les classes du domaine participantes

- Conseils et pratique
 - Réutilisez (tous les artefacts précédents)
 - "Programmez"
 - Soyez précis comme si vous codiez
 - Respectez les flots de contrôle
 - Respectez le pattern MVC
- Conseil de modélisation (UML)
 - Diag. Séquence **Détaillé**
 - Stéréotypez les objets
 - Messages synchrones / <<create>> / <<destroy>>
 - Diag. Classes **Participantes**
 - Classes / attributs / méthodes
 - Associations / cardinalités / rôles
 - Généralisation
- Règles de forme valide
 - [Moodle](#)

La réalisation illustrée en 2 croquis



A8 - Factoriser & refactorer

1. A8-S1 Factoriser les éléments communs à plusieurs DCP
 2. A8-S2 Identifier les anti-patterns
 3. A8-S3 Refactorer avec design patterns
- Conseils et pratique
 -
 - Conseil de modélisation (UML)
 -

A9 - Planter

1. A9-S1 Générer automatiquement le code

ou

1. A9-S1' Faire planter le modèle ;-)

- Conseils et pratique
 -
- Conseil de modélisation (UML)
 -