

TD1 - Conception en UML

Septembre 2024

Le sujet est écrit comme [une vraie étude de cas](#). Il vous appartient de poser le vocabulaire, de poser des questions, bref de concevoir votre projet.

Dans la suite de cet énoncé, les références sont faites relativement à l'étude de cas.

Guide de Lecture de la Feuille de TD

- ⇒ Pour vous aider à identifier les points qui font l'objet d'un rendu nous utilisons la notation **Rendu**.
- ⇒ Les consignes précédées d'un □ sont là pour vous aider à identifier ce que vous devez faire et avez fait.

Mais attention, vous devez vous approprier les sujets et vous auto-organiser !

Conseils : Tout ce qui est fait en TD, vous conduit à un rendu. Si vous ne démarrez pas tout de suite, vous n'aurez peut-être pas assez de nuits pour rattraper le retard. Pour être vraiment efficaces, vous travaillez au maximum en TD. Vous préparez vos questions. Vous utilisez vos enseignants **pour valider votre travail**, pas pour le faire à votre place, ils n'en ont de toutes les façons pas du tout le temps !

Rendu 1. Formez et organisez votre équipe [selon les règles données et les enregistrer sous Moodle](#). Tous les membres de l'équipe s'enregistrent évidemment.

1. Objectifs du TD

Dans cette première séance, une grande partie du travail consiste à analyser et comprendre le sujet. De la même manière que le client n'est pas disponible en permanence, vos enseignants ne pourront pas être présents de manière continue.

- Mettez-vous d'accord sur la compréhension du projet, au travers de la modélisation UML.
- Ce n'est pas urgent, mais réfléchissez également au rôle pris par chaque membre de l'équipe (P1) et explications dans l'annexe de l'étude de cas. Vous le préciserez également dans la définition des groupes.
- Préparer vos questions ;
- Sauvegarder votre travail parce que vous travaillerez dessus jusqu'à la fin du projet.

Ne dessinez pas, modélisez.

2. Outils

Vous pouvez commencer par travailler avec papier et crayon.

- Choisissez un outil de modélisation UML, plusieurs vous sont donnés [ici](#).

Attention si vous n'utilisez pas d'outils dédiés vous risquez de dessiner au lieu de modéliser. Il est très vivement déconseillé d'utiliser drawIO par exemple. ON NE DESSINE PAS.

 La semaine prochaine vous aurez une [livraison A2](#) dans la foulée du TD. Ces diagrammes seront utilisés et mis à jour pendant tout le module avec une [deuxième livraison D1](#) à mi-module !

3. Travail à réaliser en équipe

3.1. Former les équipes

Rendu 2. Enregister votre équipe dans [Github classroom](#) en respectant bien la règle même nom exactement que dans Moodle.

3.2. Première analyse textuelle des spécifications

- Lisez l'étude de cas si ce n'est pas encore fait.
- Mettez-vous d'accord sur les termes qui vous posent questions, notamment dans le glossaire du domaine, en considérant uniquement le périmètre correspondant aux exigences dont l'identifiant est noté en bleu dans l'étude de cas. Dans cette étape, commencez seulement à poser les bases pour ne pas vous perdre dans les détails.
- Partagez un document commun dans lequel vous mettez à jour vos définitions. Vous pouvez passer en français si vous le souhaitez.

Rendu 3. Le glossaire restreint aux termes pertinents pour la compréhension au sein de l'équipe et avec le « potentiel » client ([livraison A2](#)).

3.3. Première modélisation

Cette première modélisation intègre : N1, N2, O1-O7, P1-P3, R2, E1.

Au fur et à mesure que le projet avancera vous intégrerez ensuite les autres exigences.

Pour l'instant, votre objectif est d'apprendre la norme UML, de la respecter, de traduire votre analyse en modèles et de vérifier la cohérence dans votre analyse du domaine. Cette cohérence est essentielle pour faciliter le développement qui va suivre.

 Vos diagrammes évolueront en fonction de vos avancées et des nouvelles fonctionnalités. Organisez-vous bien pour partager, sauvegarder et maintenir vos diagrammes.

Les étapes qui suivent sont menées à la convenance de l'équipe.

a) Grands cas d'utilisation

In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. **Erreur ! Source du renvoi introuvable.** A use case diagram specifies the actors involved, the main functionalities the system supports for these actors, and the relationships (extends, includes, and specializes) between them.

- Déterminez quels sont les acteurs
- Identifiez les grands cas d'utilisation
- Prenez bien compte de l'ensemble des spécifications pour identifier les acteurs en choisissant un nom unique, les grands cas d'utilisation et leurs relations éventuelles.
- Modélez un premier diagramme de cas d'utilisation (ne cherchez pas à faire compliqué).
- Mettez-vous d'accord et partagez un document commun, quand bien même il contiendrait des copies d'écran ou des photos de versions rédigées à la main.

Rendu 4. Diagramme de cas d'utilisation restreint au périmètre donné. ([livraison A2](#)).

b) Analyse du domaine : Diagramme de classe

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. ... You use class diagrams to model the static design view of a system. For the most part, this involves modeling the vocabulary of the system, modeling collaborations, or modeling schemas. Class diagrams are also the foundation for a couple of related diagrams: component diagrams and deployment diagrams

- Identifier les principaux concepts du domaine qui pourraient former des classes dans un diagramme de classes.
- Porter un soin particulier à définir les « relations » entre les classes. Attention toute association est définie par des rôles et des multiplicités. Sans ces informations, elle ne sert à rien !

Rappel : un diagramme de classes au niveau du domaine ne vise qu'à représenter les grands concepts et à identifier les responsabilités¹ portées par les classes. Il ne contient sûrement pas de structures telles que des tableaux, des maps etc. Cela sera décidé ultérieurement quand on aura une meilleure connaissance du projet et que nous serons en mesure de faire des choix de conception.

Rendu 5. Diagramme de classes restreint au périmètre donné. ([livraison A2](#)).

c) Analyse du domaine : Diagramme de séquence

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. The sequence diagram shows the objects and the messages between the objects. **Erreur ! Source du renvoi introuvable.**

Rappel : un diagramme de séquence fait référence à un unique cas d'utilisation.

¹ Une responsabilité est une sorte de contrat, ou d'obligation, pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. En fait, on peut dire que les attributs, les opérations, et les associations représentent les propriétés élémentaires qui contribueront à remplir les responsabilités de la classe. **Erreur ! Source du renvoi introuvable.**

- Modélez sous la forme d'un diagramme de séquence, la prise d'une commande individuelle dans le contexte d'une commande groupée dont l'heure de livraison a été présélectionnée :
 - Identifiez bien les exigences que vous devez prendre en compte avant de commencer
 - Ne gérez pas les cas d'erreurs mais mettez des notes pour ne pas les oublier.
 - Ne modélez pas d'autre use case dans le diagramme de séquence ; certains sont des préconditions (le client est déjà connecté par exemple).
- Prendre un autre cas d'utilisation que vous considérez comme important et modélez un diagramme de séquences correspondant sans gérer les cas d'erreur. Vous pouvez cependant les faire apparaître par des notes.
- Assurez-vous de la cohérence entre votre diagramme de séquence et votre diagramme de classe, que vous mettez éventuellement à jour.

Dans ces diagrammes vous cherchez la cohérence entre votre analyse du domaine et le comportement réel du système (essayez de vous mettre à la place de l'acteur).

Rendu 6. *Diagramme de séquence restreint à la prise de commande dans le contexte d'une commande groupée. Vous ne devez pas faire apparaître d'objet : System ou BD ! ([livraison A2](#)).*

d) Maquette

- Produisez une esquisse d'interface(s) graphique(s) correspondant au diagramme de séquence précédent. Votre objectif est de vous assurer que vous avez bien
 - capturé le métier (Quels sont les objets nécessaires au scenario ?)
 - répondu aux exigences (Quelles sont celles que vous couvrez ?)

Rendu 7. *Esquisse de l'interface graphique restreinte à la prise de commande dans le contexte d'une commande groupée. Ne vous trompez pas, on vous demande une esquisse parce que l'interface en elle-même n'est absolument pas l'objectif de ce rendu. ([livraison A2](#)).*

e) Glossaire

- Vous mettez à jour votre glossaire, en donnant vos propres définitions du vocabulaire vraiment utile pour un réel “shared understanding of key concepts and terminology among stakeholders and team members” 😊.

N'oubliez pas de clarifier les rôles des acteurs en relation avec le diagramme de cas d'utilisation.