

Architecture Applicative

TP : Analyse de l'Architecture

1. Objectif du TP

Ce TP a pour but de définir l'architecture applicative du système d'étude du contrôleur d'un ascenseur. Ce système est présenté dans le document « Etude_Cas-ascenseur.pdf ».

Cette architecture applicative couvre le système de contrôle, mais aussi l'environnement de simulation permettant de faire fonctionner le système de contrôle et de l'étudier. Pour définir cette architecture applicative, on fera une hypothèse de structuration en définissant différents sous-systèmes.

Pour préciser le rôle des différents sous-systèmes, on utilisera le formalisme des Use Case Diagram. Pour définir les interfaces, on utilisera celui des Class Diagram. En dehors de l'application des méthodes vues en cours, ce TP nécessite de bien comprendre le problème posé, cet effort d'abstraction est la principale difficulté de l'architecture applicative. En pratique, quelle que soit la qualité des architectes applicatifs, il est rare d'obtenir une architecture définitive à la première tentative. Il faut plutôt voir l'architecture applicative comme une hypothèse de travail pour développer l'analyse d'un problème. Cette hypothèse est particulièrement importante, car elle permet l'analyse en parallèle. Cette analyse permettra de valider l'architecture applicative afin de développer un système sur une architecture stable.

Lors de ce TP chaque binôme développera sa propre architecture applicative. Pour les étudiants alternant, le travail s'arrêtera là. Pour les étudiants classiques, lors de l'UCE application, les étudiants se constitueront en équipe de plusieurs binômes, un binôme jouera le rôle d'intégrateur, tandis que les autres développeront chacun l'analyse d'une partie.

2. Déroulement des TP et évaluation

La dernière séance de TP sera consacrée à la finalisation de votre TP et à son rendu.

- partiel écrit durée :1h; coeff UCE 1/2
- rendu de TP; coeff UCE 1/2

3.

4. Déroulement du TP

4.1. Structuration du projet

Pour réaliser l'architecture applicative, vous allez créer un projet Papyrus TP-AA-Papyrus-nom1-nom2 où nom1 et nom2 sont les noms des étudiants du binôme. Vous créerez dans votre projet 2 modèles : la « use case view » et la « logical view ». Dans la « use case view », vous définirez les parties du système à partir des use case diagrams. Dans la « use case view », vous définirez les interfaces des sous systèmes et leurs interactions à partir des class diagrams.

4.2. Réaliser le use case du système.

Pour réaliser une architecture applicative, il faut être sûr d'avoir bien défini le système à réaliser. Vous devez donc, à partir de l'état du problème réaliser la phase de use case.

C'est la partie la plus délicate de votre TP, car l'état du problème vous présente les besoins de l'utilisateur, mais ne définit absolument pas le système à réaliser.

Dans cette phase, vous réaliserez :

- 1) **La vocation du système** : dans la documentation du paquetage correspondant au système vous décrirez la vocation du système.

Pour vous aider à réaliser cette tâche, nous vous donnons un questionnaire dont vos réponses vous permettront de prendre les bonnes décisions pour la définition du système.

Quel est la vocation du système (un texte court qui définit cette vocation) ?

Est-ce que le nom de votre système représente bien cette vocation.

- 2) **Le use case diagram** : Les diagrammes du système doivent être réalisés dans le paquetage "fr.univ-avignon.m1ilsen.aa.system_name" de la use case view, où system_name est le nom que vous avez donné à votre système (un nom en anglais est toujours mieux).

- 3) **Les sequence diagram pour chaque use case que vous avez identifié (optionnel).**

- 4) **La spécification fonctionnelle de l'interface utilisateur de la simulation** : dans la documentation du paquetage correspondant au système. Cette spécification reprend les use case identifiés dans le use case diagramme, mais les présente sous la forme d'un texte en français.

4.3. Réalisez l'architecture applicative :

L'architecture applicative se réalise en deux temps : la définition des parties, la définition des interfaces.

4.3.1. La définition des parties :

Comme il vous est expliqué dans le cours, pour réaliser la définition des parties, il faut que vous fassiez une hypothèse de structuration. À partir de cette hypothèse, vous créerez, dans le paquetage du système "fr.univ-avignon.m1ilsen.aa.system_name" de la use case view, autant de paquetages que de parties. Cette hypothèse se représente dans le paquetage du système par un package diagram qui représentent les paquetages des sous-systèmes et leurs dépendances. Vous réaliserez les diagrammes d'un sous-système (partie) dans le paquetage de ce sous-système. Depuis un sous-système, les autres sous-systèmes sont vus comme des acteurs qui sont créés dans le paquetage du sous-système. Par exemple, si le sous-système S1 est vu dans les sous-systèmes S2 et S3, il y aura 2 acteurs S1 : un dans les paquetages du sous-système S2, un autre dans le paquetage du sous-système S3.

Vous devez faire :

- 1) **Le package diagram du système.**

Pour chaque sous-système :

- 2) **La vocation du sous-système** : dans la documentation du paquetage du sous-système.
- 3) **Le use case diagram du sous-système.**
- 4) **Le sequence diagram de chaque use case du sous-système (optionnel).**

Avant de commencer les diagrammes des sous-systèmes, nous vous recommandons d'écrire les vocations de tous les sous-systèmes.

4.3.2. La définition des interfaces :

La définition des interfaces se fait dans le diagramme de classe, il permettra de réaliser l'architecture applicative qui sera à la base de l'analyse des parties.

Comme il vous est présenté dans le cours, vous devrez créer, dans le paquetage du système "fr.univ-avignon.mlilsen.aa.system_name" de la logical view, les paquetages de chacun des sous-systèmes. Dans chaque paquetage, vous créerez un paquetage pour l'implémentation et un paquetage pour l'interface. Chaque interface doit être décrite précisément, afin d'avoir un environnement précis pour réaliser l'analyse des sous-systèmes. Vous résumerez votre architecture dans un package diagramme dans lequel vous figurerez les paquetages d'implémentation et les interfaces qui les relient.

Vous devez faire :

- 1) **La description précise des interfaces** : sous la forme d'un class diagram dans chaque paquetage interface.
- 2) **Le package diagram du système** : sous la forme d'un class diagram dans le paquetage du système.
- 3) **Le sequence diagram de chaque use case du sous-système (optionel).**
- 4) **Le sequence diagram de chaque use case du système (optionel).**