

Projet Tuteuré
2014/2015



Création d'un logiciel de gestion de parc informatique



Caffy Cédric
Mazel Victor
Plas Julien
Traineau Kevin
Tricard Thibault

Table des matières

Introduction	3
La démarche adoptée	3
1. Diagramme Pieuvre	3
2. Diagramme FAST.....	3
3. Le Modèle Conceptuel de Données (MCD)	3
4. Maquettes	4
5. Implémentation.....	4
La gestion du projet	5
L'avancement du projet.....	6

Introduction

Nous devons, dans le cadre de notre projet tuteuré, réaliser une application d'aide à la gestion d'un parc informatique pour la société ADAM SAS. Le client, M. Vancampen, souhaite un logiciel lui permettant d'avoir un état de son parc informatique en temps réel. Son parc informatique est composé de 8 sites situés à des lieux géographiques différents. Il souhaite pouvoir accéder aux informations d'une machine en « maximum 3 clics » et veut pouvoir faire un bilan de son parc informatique sous forme d'un tableau récapitulatif.

La démarche adoptée

1. Diagramme Pieuvre

Il nous a fallu tout d'abord isoler les différentes fonctions les unes des autres et établir des ordres de priorités. Pour arriver à nos fins, nous avons utilisé le diagramme pieuvre, faisant partie de la méthode APTE. Cela permet de définir des fonctions, des acteurs et ce qu'on peut considérer comme un environnement. Une fois ce diagramme réalisé, il a fallu, définir quelles seraient les fonctions principales et quelles seraient les fonctions contraintes. Les fonctions principales correspondent aux objectifs des différentes relations créées par l'objet avec d'autres éléments extérieurs, et les fonctions contraintes (secondaires) correspondent aux conditions d'un élément extérieur astreignant.

2. Diagramme FAST

Une fois le diagramme Pieuvre terminé et les différentes fonctions déterminées et isolées, nous avons procédé à un diagramme FAST (Function Analysis System Technique) afin de déterminer l'ensemble des fonctions techniques et des solutions envisagées. Cette méthode permet en effet, de définir les différentes fonctions techniques, qui permettent à l'objet de réaliser l'action de son utilité.

Cela nous a permis d'avoir un meilleur aperçu des différentes tâches à réaliser tout au long de la phase d'implémentation. Cela nous a aussi permis de réfléchir plus en profondeur sur les différentes solutions que nous devons choisir afin de répondre au cahier des charges défini par M. Vancampen.

3. Le Modèle Conceptuel de Données (MCD)

Suite au diagramme FAST, nous avons réalisé un dictionnaire de données afin de mettre en valeur les différentes données dont nous allons avoir besoin pour réaliser notre application.

Suite à ce dictionnaire de données, nous avons créé un MCD permettant de regrouper les différentes données et les mettre en relation. Ce MCD va aussi nous permettre de réaliser notre base de données en le transformant en Modèle Physique de Données (MPD).

4. Maquettes

Nous avons ensuite élaboré des maquettes de notre application afin que celles-ci soient validées par M.Vancampen. Nous avons pour cela utilisé le cahier des charges fourni ainsi que les différentes fonctions et solutions déterminées par le diagramme FAST et Pieuvre. L'utilisation de maquette permet une meilleure communication entre le client et l'équipe de développement, car très rapidement le client peut voir si c'est ce qu'il attendait ou non. L'équipe de développement peut donc modifier certains points avant de rentrer dans sa phase de développement. Cela évite donc de perdre du temps en rendant une application non conforme aux souhaits du client, et donc de devoir recommencer la phase de développement.

5. Implémentation

Nous souhaitons utiliser le design pattern « MVC » afin d'organiser la gestion des interfaces graphiques avec les classes métiers du programme. Nous avons également choisi d'utiliser un design pattern « adapter » pour faire la correspondance JAVA-SQL. En effet, si le client souhaite pouvoir utiliser SQL Server plutôt que MYSQL, il pourra le faire facilement en n'implémentant que quelques méthodes et n'aura pas besoin de modifier tout le code qui gère les BD.

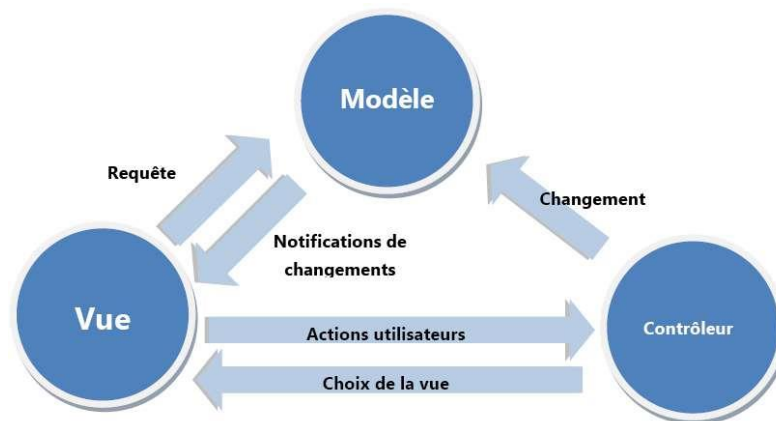


Figure 2 : Le pattern MVC

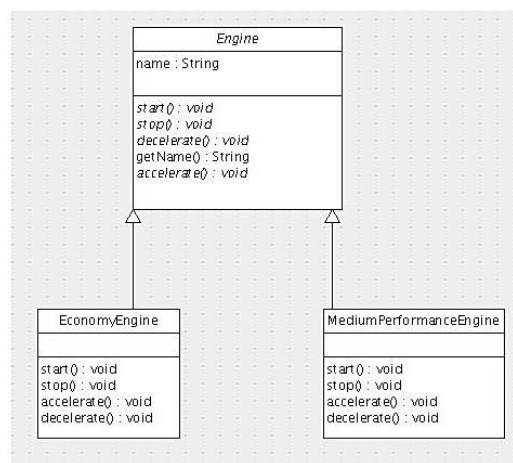
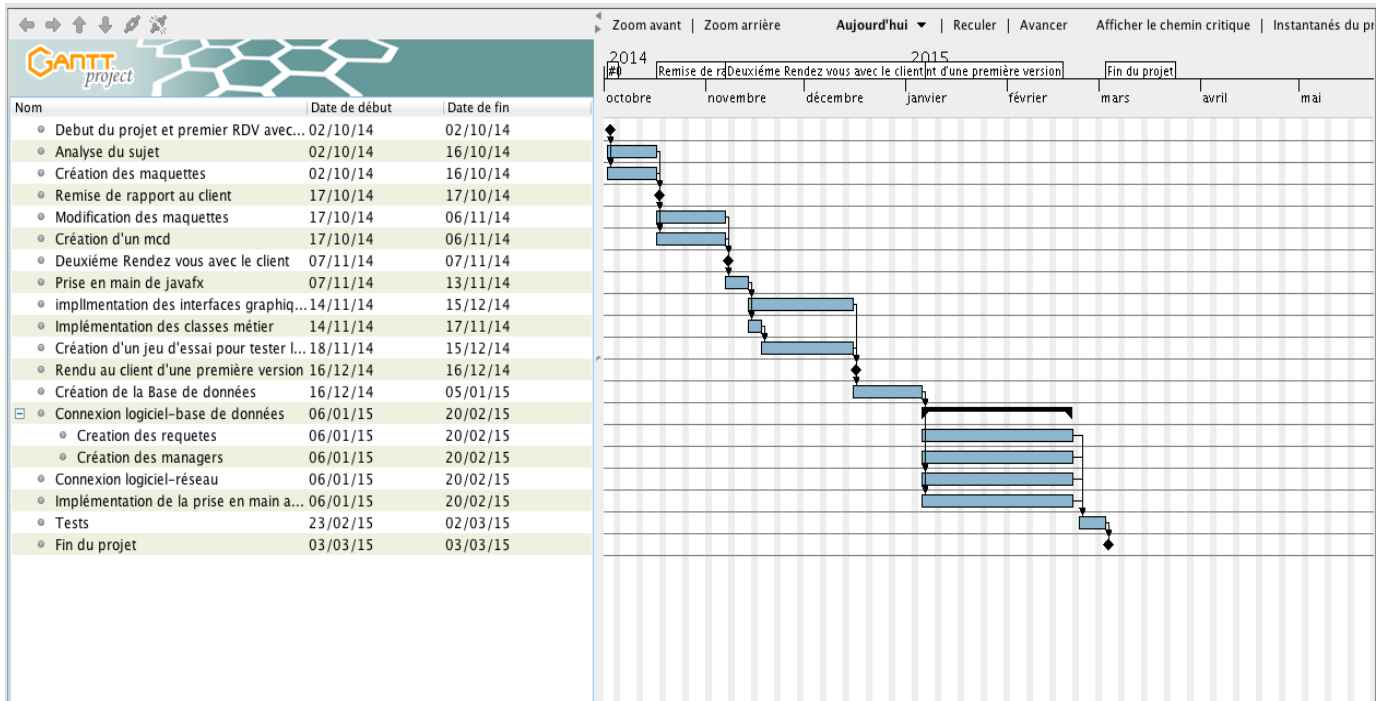


Figure 1 : Le pattern Adapter

Le design pattern adaptateur permet d'utiliser plusieurs moteurs en réécrivant les méthodes `start()`, `stop()`, `accelerate()`, `decelerate()`.

La gestion du projet

Une fois l'analyse terminée et les différentes tâches dégagées nous avons pu prioriser les tâches et estimer leurs durées. Voici le diagramme de Gantt qui en résulte.



Pour la première partie du projet (de la première réunion à la deuxième) nous avons travaillé l'analyse lors de réunions.

En revanche après la deuxième réunion nous avons réparti les tâches comme ceci :

Tâches	Personnes Affectées
Prise en main de javaFx	Tout le monde (séparément)
Implémentation des interfaces graphiques	Cédric, Victor, Julien, Kévin
Implémentation des classes métier	Thibault
Création d'un jeu d'essais	Thibault
Création de la base de donnée	Tout Le monde
Connexion logiciel-BD	Thibault, Cédric
Connexion logiciel-réseau	Julien
Implémentation prise en main à distance	Kévin, Victor
Test	Tout le monde

L'avancement du projet

Après cette phase d'analyse, nous avons décidé du langage de programmation que nous utiliserons en fonction des demandes techniques du projet et des différents langages.

Au final, nous avons choisi, pour développer notre logiciel, d'utiliser le langage JAVA. Tout d'abord car il permet de créer des interfaces graphiques assez facilement. Nous avons également préféré le JAVA à d'autres langages pour sa sécurité, car étant très typé, stable, documenté avec une forte communauté, et surtout portable.

Pour réaliser nos interfaces graphiques, nous avons choisi d'utiliser la bibliothèque « JavaFX » de JAVA, plus récente et plus souple que l'ancienne bibliothèque de création d'interface de Java, « Java Swing ».

De plus, nous avons utilisé le logiciel SceneBuilder, qui permet de créer une interface en JavaFX, notamment grâce à des « Drag N Drop » des différents composants à insérer sur la page.

A ce jour, nous avons créé toutes nos classes métiers, et nous terminons les classes permettant de gérer l'interface graphique. De plus, nous avons réalisé un jeu d'essai afin de pouvoir tester l'interface graphique du logiciel.

Nous nous sommes mis d'accord avec le client, M. Vancampen, de rendre pour le lundi 15 décembre. Cette première version du logiciel, sans fonction implémentée, mais avec une interface fonctionnelle et des jeux d'essais concrets, afin que d'éventuelles modifications ou adaptations de nos interfaces puissent être effectuées avant de commencer l'implémentation des fonctionnalités.