ASI

Ricardo CORREIA | Liasse LOPES

Programmation Orientée Objet

Report from Canvas

OBJECTIFS :

L'objectif de le project est de créer des estimateurs, qui, à partir de données récoltées par des capteurs, soient capable de déterminer le nombre de personnes dans le bureau de la figure 1. Le nom de les capteurs apparaît en vert. Pour entendre meilleur le code un javadoc a été généré et est accessible via \Project Final\dist\javadoc\index.html

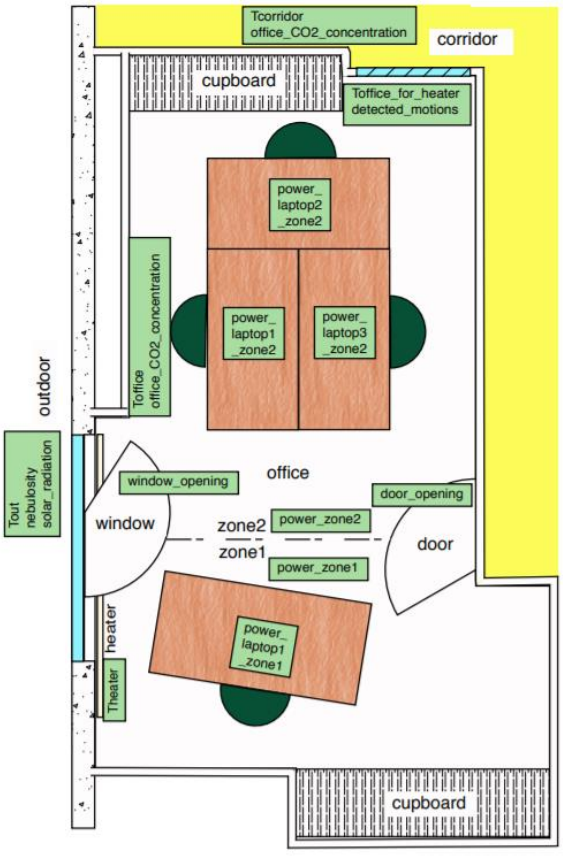

Figure 1 - Plan of the office under consideration

Sommaire

ARCHITECTURE

Les classes d'objets ont été codées de la manière la plus générale possible, c'est-à-dire aussi indépendantes que possible des autres classes, dans un souci de lisibilité, mais aussi dans un souci de réutilisation.

En conséquence, nos classes principales sont presque vides (généralement quelque chose comme la nouvelle classe ()) et nos fichiers de classe sont petits (200 lignes maximum, mais plus petit est généralement préférable).

La figure 2 montre le diagramme de classes UML du projet.

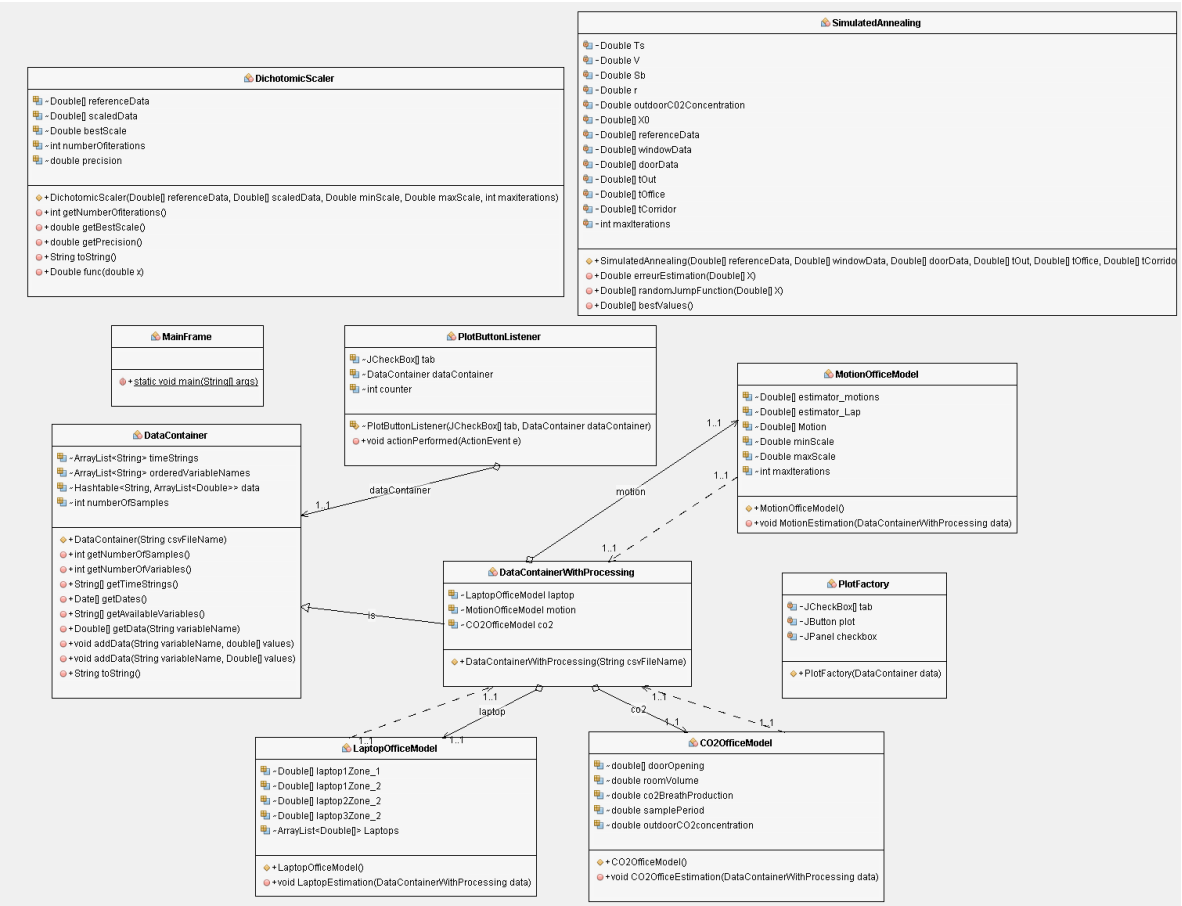

Figure 2 - UML Class Diagram

Notre projet est divisé en trois « packages », le package *data* est destiné aux traitements des données des capteurs, et les calculs des estimateurs. Le package *gui*, s'occupe de l'interface graphique permettant de tracer les courbes des données et les résultats des estimateurs, et contient aussi la classe « main » du projet, et, le package *fitters* s'occupe de faire les calculs d'estimation.

PACKAGE GUI

MainFrame.java

La classe MainFrame permet de lancer le projet.

PlotFactory.java

La classe Plotter permet de créer la fenêtre contenant qui permet de sélectionner les différentes courbes que l'on souhaite tracer et un bouton que permettant de lancer le tracer des courbes.

C'est une extension de la classe JFrame qui contient deux JPanel, un avec un tableau de JCheckBox qui contient le nom des variables ainsi que les résultats des estimateurs, et l'autre avec les boutons décris ci-dessus. Ceux-ci sont associés à deux « écouteurs » qui permettent d'exécuter les actions souhaitées.

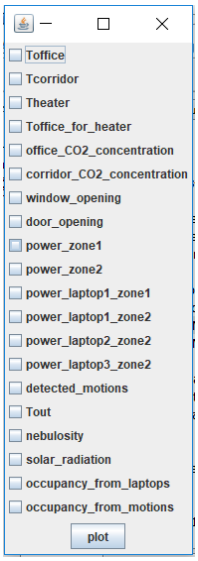

Figure 3 - GUI Initial

PlotButtonListener.java

Cette classe est responsable de la gestion du bouton de tracé, elle fonctionne avec un écouteur d'action qui vérifie quelles données et quels estimateurs doivent être tracés.

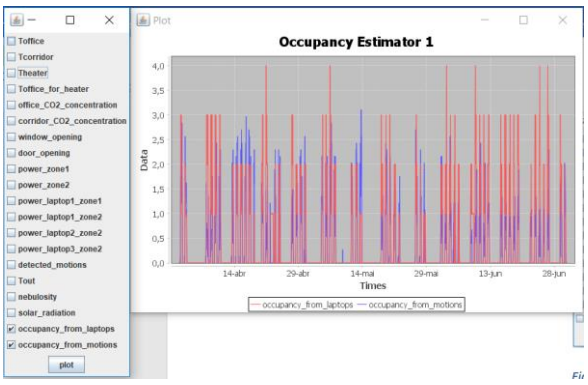

Figure 4 - GUI avec plot

PACKAGE DATA

DataContainer.java

Les classes « DataContainer » qui permet de stocker et traiter les données des différents capteurs et « ReadCSV » qui permet de lire les données du fichier Excel utilisées dans le DataContainer nous ont été données, nous allons donc développer les autres classes et leurs fonctionnements.

DataContainersWithProcessing.java

Cette classe est une extension de DataContainers, elle contient en plus, les objets permettant d'obtenir l'estimation du nombre de personne par deux méthodes différentes et les stockent.

SimulatedAnneling.java

Cette classe est responsable de l'exécution de calculs dans un algorithme d'optimisation d'amélioration multidimensionnelle avec diversification pour l'estimateur de CO2. Malheureusement, cet estimateur ne fonctionne pas.

PACKAGE FITTERS

LaptopOfficeModel.java

Cette classe est responsible pour le premier estimateur, on utilise une simple comparaison avec une boucle « if », et on ajoute la somme des estimations dans une nouvelle variable. La consommation de chacun des 4 ordinateurs portables est mesurée par un wattmètre (variable 'power_laptopX_zoneY'). Si la consommation moyenne est supérieure à la consommation en veille (15W), on considère que quelqu'un travaille sur l'ordinateur.

MotionOfficeModel

Dans cette classe on utilise les résultats du premier estimateur (power laptop) pour obtenir la valeur du coefficient avec la classe DichotmicScaler présentée ci-dessous. On multiplie ensuite le coefficient avec les données du capteur de mouvement et obtenir ainsi une estimation du nombre de personne.

DichotomicScaler.java

Classe à l'échelle à juste titre l'occupation basée sur le mouvement. Afin de déterminer le meilleur coefficient proportionnel, une dichotomie simple peut être réalisée, en considérant les estimations basées sur la consommation d'ordinateurs portables comme les occupations réelles. Échelle le coefficient pour minimiser l'erreur entre la consommation d'ordinateurs portables et les estimateurs d'occupation basés sur la détection de mouvement.

CO2OfficeModel.java

Cette classe est responsable de l'exécution de calculs dans un algorithme d'optimisation d'amélioration multidimensionnelle avec diversification pour l'estimateur de CO2.
Malheureusement, cet estimateur ne fonctionne pas.

RÉSULTAT ET CONCLUSION

Pour commancer, les considérations pratiques doivent être prises en compte. Notez qu'un compteur de puissance en communication coûte environ 80 euros, un détecteur de mouvement à 120 euros, un capteur de contact pour porte ou fenêtre 50 euros et un capteur de CO2 environ 250 euros. Selon la figure 1, nous avons quatre capteurs de puissance d'un coût total de 320 euros. Nous avons également deux capteurs de mouvement, chacun au coût de 120, pour un total de 240 euros. Enfin, les capteurs de porte et de fenêtre ainsi que les capteurs de CO2 ont un coût total de 600 euros à mettre en œuvre.

Il convient également de rappeler que pour effectuer une estimation correcte par mouvement, il est nécessaire de trouver le coefficient proportionnel basé sur la consommation des ordinateurs portables, c'est-à-dire d'implanter l'estimation par mouvement, nous avons besoin des capteurs de consommation d'énergie.

En résumé, nous aurons les coûts suivants pour la mise en œuvre des systèmes d'estimation (tableau 1).

Estimation par	Nombre de capteurs	Valeur totale
Power Consumption	4	€ 320.00
Motion	6	€ 560.00
CO2	4	€ 600.00

Tableau 1 - Coûts totaux

Malheureusement, nous n'avons pas pu coder l'estimation de l'occupation basée sur le CO2. Cette solution a donc été abandonnée en raison de la complexité et du coût.

Par contre, il est difficile d'analyser les graphiques d'estimation basés sur la consommation et le mouvement (figure 5 et 6) pour savoir lequel était le plus précis dans l'estimation, car nous n'avons pas d'informations concrètes sur l'occupation réelle heure par heure du bureau.

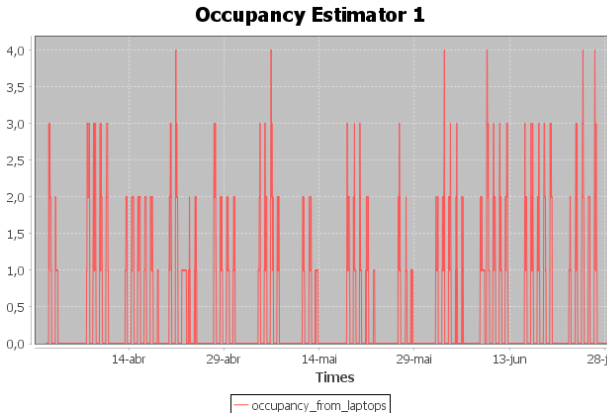

Figure 5 - Estimation par consommation des laptops

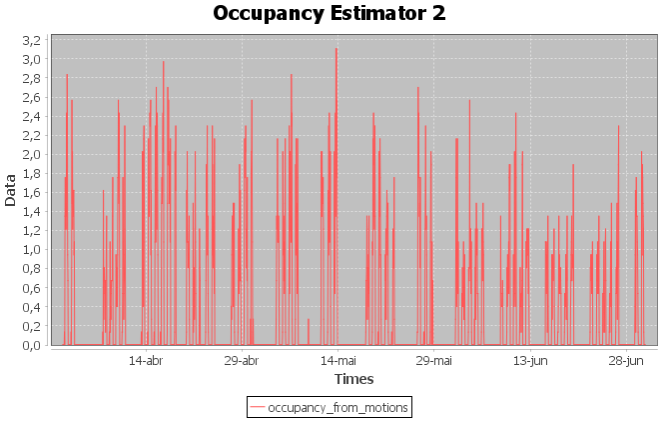

Figure 6 - Estimation par Motion

De cette manière, nous pensons à utiliser le méthode de consommation des ordinateurs portables pour estimer l'occupation du bureau afin d'avoir le coût le moins cher, être plus simple et plus facile à mettre en œuvre.