

TD n°3

Le Modèle-Vue-Contrôleur (MVC)

1 Rappel

Le modèle MVC (Model-View-Controller) consiste à découpler une application en trois principaux rôles (le modèle, la vue et le contrôleur), afin que chaque rôle puisse être modifié ou remplacé facilement, et indépendamment des autres.

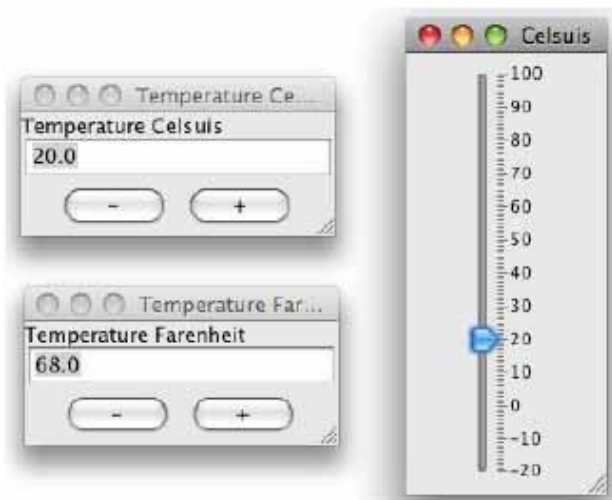
- Le **modèle** représente les données manipulées par l'application et il offre des méthodes d'accès et de modification de ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.
- Le **la vue** correspond à l'interface avec laquelle l'utilisateur interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.
 - Affiche le contenu du modèle (accède aux données dans le modèle)
 - Spécifie comment ces données se visualisent
 - Lorsqu'il y a des changements au modèle, la vue doit maintenir la cohérence de la représentation du modèle
- Le **contrôleur** prend en charge la gestion des événements pour mettre à jour le modèle. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle.
 - Fait la traduction entre **une action faite par l'utilisateur** dans la vue et une action à faire dans le modèle.

Le patron MVC (Modèle-Vue-Contrôleur) peut être vu comme une combinaison des trois patrons **composite**, **observer** et **strategy** :

- S'il y a plusieurs vues, elles peuvent être organisées dans un arbre en utilisant le patron « composite »
- La relation entre les vues et les modèles est le patron « observer ». Le **modèle** agit comme le **Sujet** et la **Vue** prend le rôle de l'objet **Observateur**.
- Les contrôleurs sont des « strategy » des vues. La **vue** est un **Contexte** et le **Contrôleur** est un objet **Stratégie**. **On dit aussi que le contrôleur est la strategy de la vue**. Le contrôleur va transférer les données de l'utilisateur au modèle et il a tout à fait le droit de modifier le contenu.

2 Exercice

L'objectif est de créer une application permettant le contrôle d'une température en degrés Celsius ou Fahrenheit. L'interface se compose de trois vues représentant la même température sous des formes différentes. La modification d'une des vues doit mettre automatiquement à jour les autres vues.



Question : Proposez une modélisation UML de cette application avec le patron MVC.