

Illustration de la conception d'architecture logicielles :

Gestion de robot peintre pour voiture

Description de l'application

Nous considérons une partie de la chaîne de production d'une voiture : la salle de peinture des voitures.

Cette salle est constituée de robots peintres : chaque robot doit être nettoyé soit lorsque la couleur change, soit suffisamment régulièrement pour son entretien.

L'application vise à assister la gestion de cette salle de peinture et la salle de nettoyage de robot en permettant de :

- attribuer un robot à la peinture d'une voiture
- demander le nettoyage d'un robot
- récupérer des informations sur les robots et les voitures

Spécification

On considère que l'application utilise/modifie :

- des données stockées dans une base de données
- une interface graphique
- une API qui gère le nettoyage des robots

Spécification : BD

Une voiture est définie, entre autres, par :

- numéro d'identification
- dates de début et de fin de production
- étape de production en cours

Un robot est défini, entre autres, par :

- numéro d'identification
- état : en service, au repos, hors-service, en nettoyage
- date du dernier nettoyage
- couleur courante
- identification de la voiture en cours de peinture

Spécification : Interface Graphique

Le visuel doit toujours apparaître sur une seule page. Cette page contient :

- Une liste de robots
- Une liste des voitures qui sont dans la salle de peinture
- Une fenêtre d'affichage des informations demandées
- Un bouton pour lancer le nettoyage du robot sélectionné dans la liste
- Un bouton pour lancer la peinture de la voiture sélectionnée par le robot sélectionné

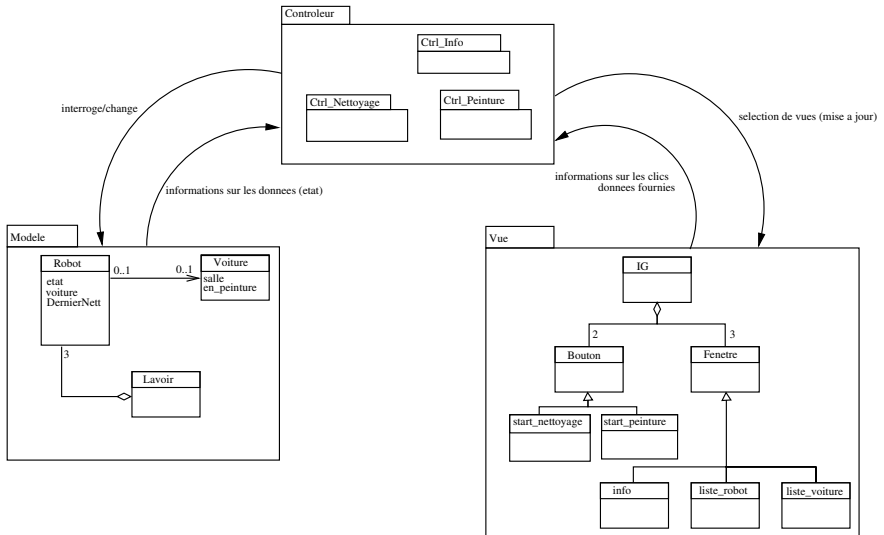
La partie "affichage" permet de visualiser :

- l'acquittement ou le problème rencontré suite au lancement d'une action (peinture ou nettoyage)
- les informations relatives au robot et à la voiture sélectionnés

Pourquoi utiliser MVC ?

- Besoin de contrôler la vue (IG) et le modèle (utilise la BD)
- Chaque clic sur la vue demande une récupération/modification du modèle qui modifiera la vue (affichage)
- La solution présentée utilise 3 contrôleurs pour l'affichage des informations, le nettoyage et la peinture

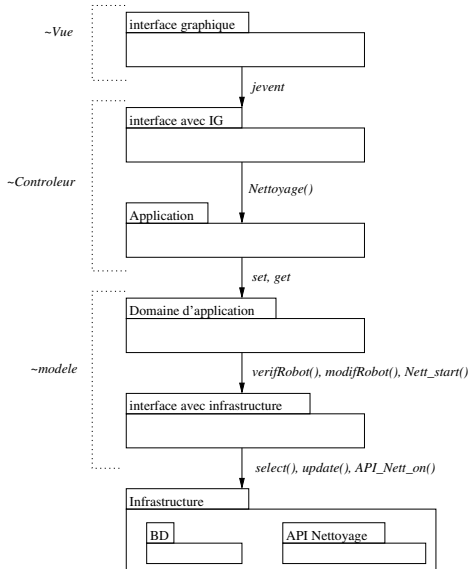
Gestion de robot peintre de voiture : MVC



Pourquoi utiliser l'architecture en couches ?

- Les différentes parties de l'application peuvent être développées indépendamment.
- Le code de l'application est isolé "par couche". Les modifications ci-dessous n'altèreraient pas le coeur de l'application :
 - ▶ L'affichage pourrait évoluer d'une interface graphique vers une page html
 - ▶ Les données pourraient être répertoriées dans un fichier à la place d'une BD
 - ▶ L'API de nettoyage pourrait évoluer

Gestion de robot peintre de voiture : MVC



Les appels de méthodes sont des exemples des méthodes appelées lors de la demande de nettoyage d'un robot.

Comment concevoir une architecture ?

L'architecture logicielle dépend de :

- type d'application :
 - ▶ besoin de contrôleur
 - ▶ taille de l'application
 - ▶ besoin d'infrastructure (exemple : BD)
- du contexte d'utilisation :
 - ▶ modularité
 - ▶ évolutivité
 - ▶ efficacité
 - ▶ isolation

Attention : une architecture peut aussi être un mélange entre une architecture "en couches" et "MVC" !