

TP 4 : Coder en Java le modèle d'usine robotisée : vue structurelle

Dominique Blouin, Télécom Paris, Institut Polytechnique de Paris

dominique.blouin@telecom-paris.fr

Le but de cet exercice est de coder en Java le diagramme de classe que vous avez dessiné lors du TD précédent. Ce diagramme a été produit alors que vous ne disposiez pas encore de toutes les notions et bonnes pratiques de modélisation qui ont vues en classe par la suite, telles que les classes abstraites et le principe de responsabilité unique par exemple. Ainsi la traduction du diagramme de classe en Java est également l'occasion d'**améliorer** votre conception en utilisant ces notions et bonnes pratiques, et de la compléter au besoin.

Comme nous l'avons vu, l'usine robotisée est inspirée du [laboratoire de systèmes cyber-physique de l'institut Hasso-Plattner](#) (Figure 1).

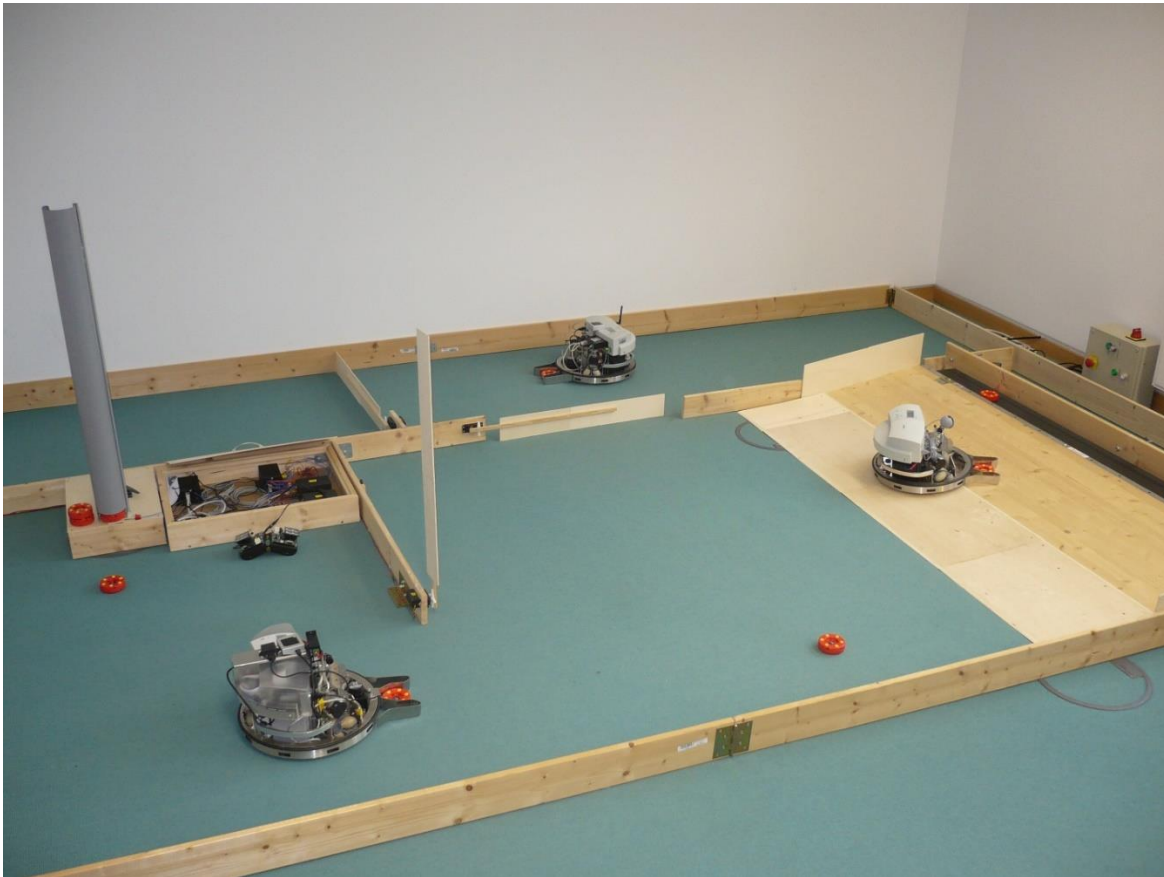


Figure 1 : L'usine robotisée du laboratoire de systèmes cyber-physiques de l'institut Hasso-Plattner.

Tel qu'illustré par le plan de la Figure 2, l'usine est constituée de différentes salles dans lesquelles des machines sont installées. Ces machines sont destinées à effectuer un traitement particulier sur des rondelles produites par l'usine.

A l'intérieur de cette usine, des robots servent à transporter les rondelles produites entre les diverses machines de production. Lorsque nécessaire, ces robots peuvent rejoindre des stations de recharge afin de recharger leurs batteries jusqu'à ce qu'ils aient suffisamment d'énergie pour reprendre leur travail.

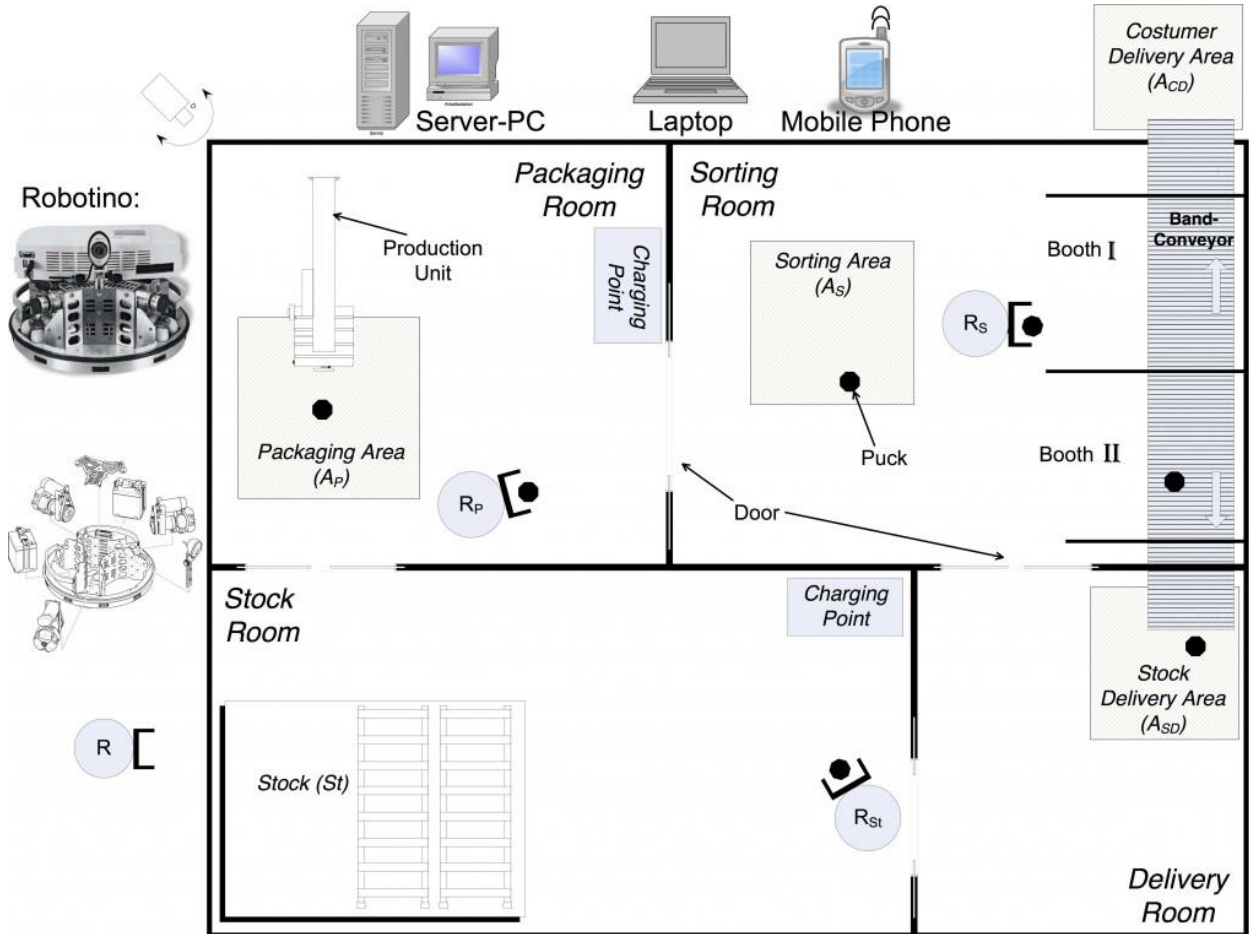


Figure 2 : Plan de l'usine robotisée.

Lors de la conception de systèmes comme une usine production robotisée, la simulation est très utile pour évaluer différentes propriétés telles que la quantité de biens pouvant être produits par unité de temps, la quantité de robots nécessaires pour transporter les biens d'une machine à l'autre, ou la capacité des batteries des robots requise pour produire cette quantité de biens, afin d'éviter que les robots ne passent trop de temps en station de recharge.

Le but du simulateur que vous allez développer pour ce projet sera de visualiser le comportement de l'usine robotisée, c'est-à-dire de visualiser différents événements tels que le déplacement des robots transportant des rondelles dans le bâtiment, l'ouverture de portes automatiques, permettant éventuellement si le temps le permet de faire quelques estimations grossières de consommation énergétique des robots. Ces derniers aspects seront vus lors de TP ultérieurs.

Les constituants du système

Nous allons d'abord modéliser les constituants de l'usine robotisée. Elle se compose de différents types de **composants** tels que :

1. Une usine.
2. Ses salles et portes incluant des aires de production.
3. Des machines de production positionnées dans les aires de production.
4. Des robots.
5. Des stations de recharge de robots.
6. Des convoyeurs.
7. Des rondelles produites par l'usine.

Créer une classe composant

Nous allons considérer que tous ces objets sont des **composants** de l'usine et que nous souhaitons représenter leur **position** dans l'usine en ne considérant que leurs coordonnées cartésiennes (position x et y) dans le plan constitué par le sol du bâtiment.

Créer une première classe nommée *Component* qui définira des attributs communs à tous les composants de l'usine tels que leur position sur le sol de l'usine et leur dimension. Au besoin, créer des classes pour ces différents aspects.

La classe *Robot* que vous avez créée lors des TP précédents devra être modifiée afin d'hériter de cette classe *Component*. Par ailleurs, modifier également la classe *Factory* afin qu'elle possède une **liste de composants** au lieu d'une liste de robots. Modifier également la méthode *addRobot()* et la renommer *addComponent()*. Cette méthode prendra en paramètre un objet de type *Component* au lieu d'un nom de robot, et elle ajoutera ce composant à la liste de composants de la classe *Factory*. Elle devra vérifier que le nom du composant passé en paramètre est unique parmi tous les noms des composants de la liste de composants de la classe.

Modéliser les différents types de composants du système

En utilisant la notion d'héritage de classe vue en cours, créer des classes pour chacun des différents types de composants de l'usine. Ajouter les attributs que vous pensez utiles pour ces différentes classes.

Redéfinir la méthode d'affichage des nouvelles classes

Redéfinir la méthode *toString()* de chaque nouvelle classe afin de personnaliser son affichage à la console.

Instancier une usine robotisée et l'afficher à la console

Afin de tester votre modèle, créer une classe test contenant une méthode *main*. Dans cette méthode, instancier une usine robotisée contenant 3 salles ayant chacune une aire de travail, contenant chacune une machine de production. Y ajouter 3 robots et une station de recharge.

Afficher cette usine à la console et vérifier que sa représentation sous forme de chaîne de caractère est correcte.