

Feuille de travaux pratiques n° 2

Projet Workshop

Etude de cas

Ce document présente le cas d'étude d'automatisation d'un atelier de production. On s'intéresse ici à deux postes de travail, un trieur et un robot manipulateur.

1 Généralités

Dans un système de production, le sous-système de décision est souvent constitué d'une couche logicielle appelée Manufacturing Execution System (MES). Par l'utilisation d'informations à jour et précises, le MES, guide, initie, réagit aux activités de l'atelier au fur et à mesure de leur déroulement, et fournit des rapports sur ces activités. Le MES vient assurer des fonctions de contrôle/commande (SCC) qui permettent le pilotage en temps réel des ateliers de fabrication.

Une ressource peut être un robot autonome ou un robot asservi à un opérateur. On utilise pour cela des outils de communication vers les automates (programmable logic controller -PLC), les robots industriels ou les AGV. Le transport est un cas particulier de ressource dans l'atelier de fabrication. La fonction de transport est alors assurée par un poste (mobile), un convoyeur ou un Véhicules à guidage automatique ou AGV (*Automated Guided Vehicles*) qui sont des véhicules sans pilote guidés par différentes technologies (laser, suivi de ligne, gps, etc.) qui déplacent les éléments (objets, produits, pièces...) d'un poste de l'atelier à un autre.

Du point de vue logiciel, pour le MES on se placera dans une approche service, ce qui facilite l'interopérabilité des systèmes et des niveaux d'abstractions (entre un processus métier et une séquence d'ordre de commande).

Par exemple, un atelier d'usinage est constitué de postes de travail de trois types, comme indiqué dans la figure 1¹. Dans cet exemple, la méthode de gestion de production, appelée Méthode du Kanban, dans laquelle la production est égale, à tout moment, à la demande. Utilisée initialement dans l'industrie, elle revient en fanfare dans le développement logiciel avec les méthodes dites agiles. Nous en décrivons ici une version considérablement simplifiée.

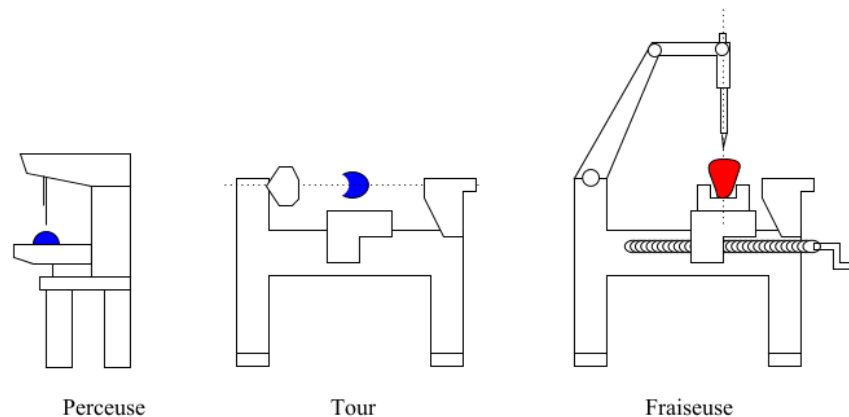


Figure 1 : Postes de l'atelier

Les pièces transitent d'un poste à l'autre pour subir divers traitements (usinage). L'ordre de transit est donné dans la figure 2.

1. page 106 de Exercices corrigés de conception logicielle, Modélisation de Systèmes d'Information par la pratique, P. André and A. Vailly, Editions Ellipses, 2003, ISBN : 2-7298-1289-X

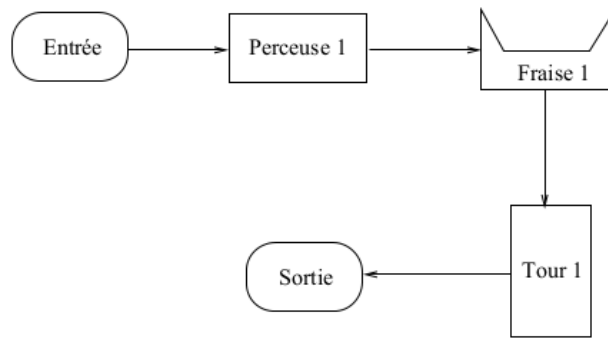


Figure 2 : Plan d'un atelier simplifié

Les pièces sont rangées dans des conteneurs. Chaque poste possède un conteneur pour les pièces à usiner. Lorsqu'un poste a usiné une pièce, il la place dans le conteneur du poste suivant comme l'illustre la figure 3.

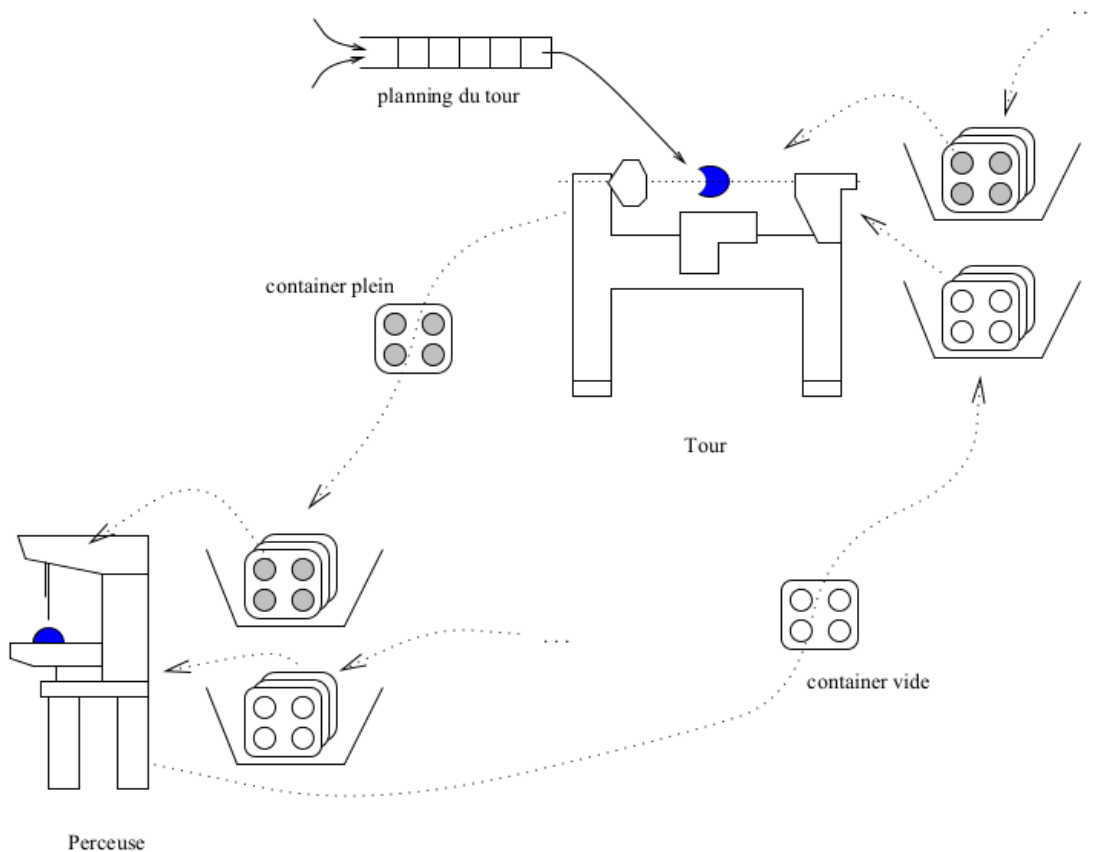
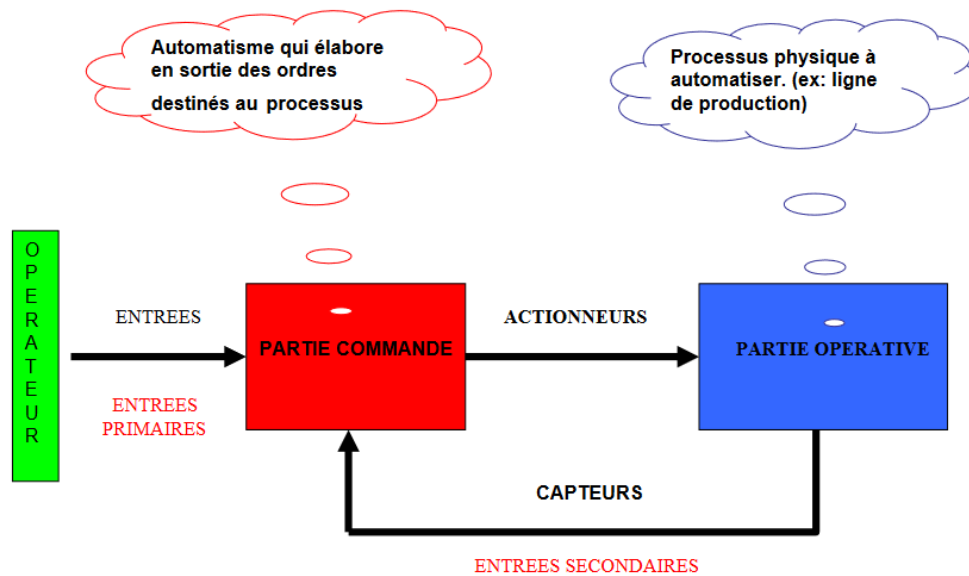


Figure 3 : Ordonnancement des opérations

Chaque ressource (physique) est un système cyber-physique ou CPS (*Cyber-Physical System*) qui comprend une partie matérielle contrôlée et commandée par une partie logicielle. Il s'agit d'automatisme, on parle aussi de systèmes embarqués ou temps réel du fait d'une réactivité forte en logiciel et matériel. En automatisme, le principe général est celui de la boucle contrôle-commande dans lequel le contrôleur reçoit les instructions, capte les événements et l'état du contexte avant de donner des ordres aux actionneurs pour agir sur le système (et donc son contexte) et ainsi de suite. Le couplage entre la partie matérielle et la partie logicielle est réalisé par ces capteurs et actionneurs.

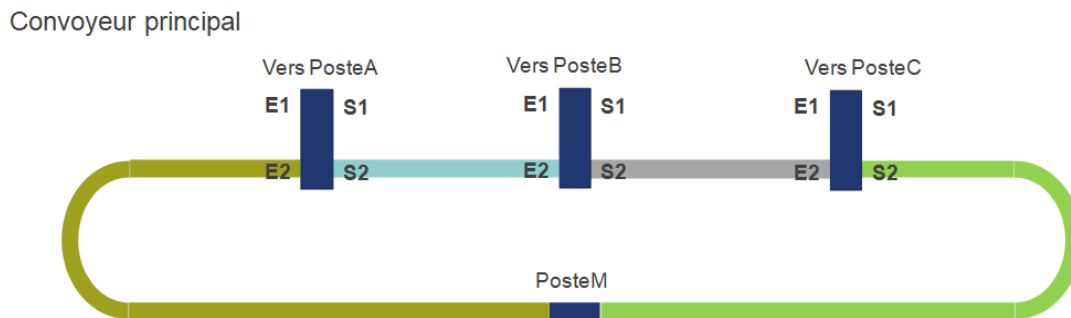


source : http://karlaoui.free.fr/SiteEpmi/Systmes_Linaires/Cours/1_Introduction_la_commande_des_systmes.pdf

Figure 4 : Boucle contrôle/commande

2 Le cas d'étude en bref

Le cas porte sur un équipement industriel simplifié (*manufacturing system*), muni de postes de travail sur lesquelles on procède à des opérations dans un ordre donné (gamme de fabrication, ordres). Pour ce type d'applications, on s'intéresse à la gestion d'atelier mais aussi la gestion individuelle des postes de travail, appelés aussi ressources. Une illustration est donnée en Figure 5.



source : Sofal

Figure 5 : Plan atelier simplifié

Nous suivons un développement itératif et incrémental (versions) des applications.

2.1 Périmètre Version 1

Dans cette itération, on se restreint à deux ressources, qu'on souhaite mettre en œuvre comme deux sous-systèmes indépendants.

- Un robot trieur de pièces (*Color sorter*) qui observe et trie les pièces fournies.
 - Un robot bras articulé pour déplacer des objets (*Robot Arm*) qui déplace des pièces.
- Chaque sous-système comprend, **dans sa version autonome**, deux dispositifs
- une interface de *pilotage* sous forme de télécommande ou de smartphone (le *client*) et
 - un système CPS (automate muni de capteurs et actionneurs branchés sur un système électro-mécanique (le *serveur*), le logiciel de contrôle du robot, qui pilote les différents dispositifs physiques.

Il s'agit de concevoir et mettre en œuvre un logiciel de gestion d'atelier, dont l'analyse du domaine et les spécifications des exigences logicielles (fonctionnelles et non fonctionnelles) sont fournies.

Dans un souci de simplification et d'évolution, nous donnons une description plus abstraite que celle du modèle d'implantation mais moins complexe que celle d'un vrai dispositif.

La mise en œuvre concrète du prototypage se fera en utilisant des automates LEGO MINDSTORMS EV3, un ensemble d'outil de construction, programmation et de commande de robots LEGO.

3 Description du fonctionnement et des composants

Les maquettes et notices de montage donnent la structure matérielle et les capteurs utilisés pour chaque sous-système. Nous complétons ici par le comportement des composants partagés (capteurs et actionneurs) et les spécificités de chaque assemblage.

3.1 Fonctionnement général

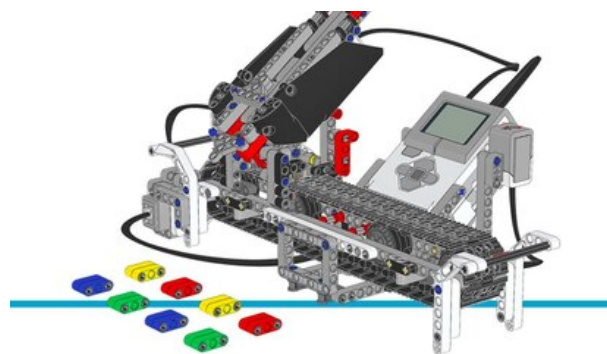
Le principe général est que l'opérateur définit la configuration et envoie les ordres de fabrication, éventuellement cet opérateur intervient dans le processus pour des opérations manuelles.

Il dispose pour cela d'une interface de pilotage qu'il doit connecter (à la demande) au contrôleur du poste (automate). En cas de perte de connexion avec l'interface de pilotage, le poste continue son activité jusqu'à la fin de l'opération, une action de l'opérateur, un dysfonctionnement ou une panne. L'opérateur devra alors relancer le processus, éventuellement en se reconnectant.

3.1.1 Trieur de couleur

Ce système comprend un lecteur de couleur, un tapis roulant mu par un moteur à rotation et borné par un capteur de contact et un système de coulissage et d'éjection mu par un moteur court de poussée. On y ajoute la notion de container en sortie. Des vidéos montrent des exemples de fonctionnement autonome, par exemple :

- <https://youtu.be/JNSZfUwDoEY>
- <https://youtu.be/QnLoZOYXsE4>
- <https://youtu.be/xOYMIy5tb70>



source : <https://ev3.univ-nantes.fr/maquettes/>

Figure 6 : Maquettes Robots Lego EV3 du trieur de pièces

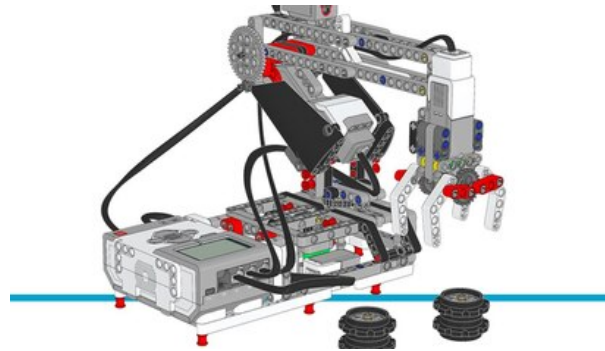
Nous souhaitons aller plus loin en proposant un paramétrage du fonctionnement par l'opérateur avec une interface sur dispositif mobile (tablette/smartphone) et la prise en compte d'informations de l'environnement. L'opérateur précisera pour chaque ordre de fabrication (ici de tri) le nombre de pièces, l'emplacement pour chaque couleur car seulement 4 bacs collecteurs sont disponibles à un moment donné. Il définira aussi le nombre d'emplacement disponibles dans les conteneurs de sortie, on ne doit pas dépasser la capacité et donc compter les capacités. Chaque conteneur est associé à une et une seule couleur.

3.1.2 Bras articulé

Ce système comprend un lecteur de couleur, une plateforme mue par un moteur à rotation et bornée par un capteur de contact, un bras articulé mue par un moteur à rotation et géré par un capteur de couleur (ou de distance) et un système de préhension mu par un moteur court de poussée.

Des vidéos montrent des exemples de fonctionnement autonome, par exemple :

- <https://youtu.be/zCELKToR7c4>
- <https://youtu.be/JKnr0FCQwII>
- <https://youtu.be/JOUGsgqYraE>
- construction <https://youtu.be/gvTyVhuV0dQ>



source : <https://ev3.univ-nantes.fr/maquettes/>

Figure 7 : Maquettes Robots Lego EV3 du bras articulé

Nous souhaitons aller plus loin en proposant un paramétrage du fonctionnement par l'opérateur avec une interface sur dispositif mobile (tablette/smartphone) et la prise en compte d'informations de l'environnement. L'opérateur précisera pour chaque ordre de fabrication (ici de déplacement) le nombre et la taille des pièces à traiter, l'emplacement des pièces au départ et à l'arrivée (on vérifiera qu'il n'y a pas deux pièces au même endroit que ce soit au départ ou à l'arrivée).

3.2 Interface de pilotage

L'interface de pilotage permet de démarrer ou arrêter le système, rend compte de l'état du système complet (poste et pièces) tel que perçu par l'automate et alertera l'opérateur pour des opérations manuelles à valider (par exemple changer de bac collecteur, ajouter, déplacer des pièces) ou en cas de dysfonctionnement du poste ou du processus en cours.

L'analyse devra fournir le comportement de l'interface (diagramme état-transition) spécifique au poste de travail traité.

La conception générale en donnera une maquette visuelle.

Une gamme de fabrication est composée d'opérations, elles-même composée de tâches élémentaires. L'interface de pilotage doit permettre de décrire des gammes de fabrication et transmettre les ordres de fabrication aux postes de travail. Un ordre de fabrication est associé à une tâche. Une gamme de fabrication est composée d'au moins une opération et une opération comprend au moins une tâche élémentaire. Il existe des opérations de configuration, contrôle, de maintenance ou même d'attentes.

3.3 Capteurs

Différents types de capteurs sont exploitables.

3.3.1 Capteur de contact

Le capteur signale à son contrôleur lorsque le contact a lieu. Il se comporte donc de façon comparable à la télécommande et nous pouvons modéliser son comportement par la figure 8. S'il y a plusieurs capteurs de contacts, on les distingue par leur identifiant, fourni au contrôleur par la variable `self`.

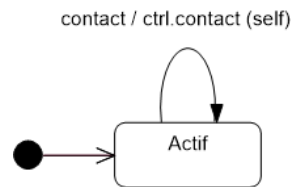


Figure 8 : Diagramme Etats-transitions du capteur de contact

3.3.2 Capteur de présence

Le capteur signale à son contrôleur lorsque le contact a lieu. Il se comporte donc de façon comparable à la télécommande et nous pouvons modéliser son comportement par la figure 9.

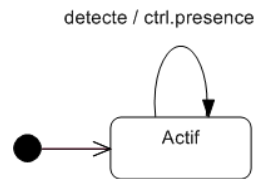


Figure 9 : Diagramme Etats-transitions du capteur de présence

3.3.3 Capteur de couleur

Le capteur mesure la couleur de la surface observée et la fournit à son contrôleur à la demande. Il se comporte donc de façon comparable au capteur de présence (distance) et nous pouvons modéliser son comportement par la figure 10. Le capteur identifie 8 couleurs distinctes : NONE, BLACK, BLUE, GREEN, YELLOW, RED, WHITE, BROWN².

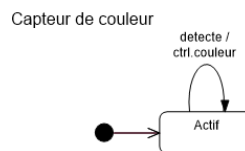


Figure 10 : Diagramme Etats-transitions du capteur de couleur

3.4 Actionneurs

Les actionneurs sont les moteurs, les sons et les affichages de console.

3.4.1 Moteur

Un moteur générique **actionne** un dispositif de mouvement. Soit il pousse, soit il tire... soit il est à l'arrêt. Nous avons donc trois états, En poussée, En tirée, Arrêt. Le résultat de ces actions est fonction du dispositif actionné : une translation pour un chariot ou une porte coulissante, une rotation pour un battant de porte, une chaîne fermée, un tapis roulant... Lorsque le moteur pousse, l'élément va dans un sens *e.g.* la porte s'ouvre ; lorsqu'il tire, l'élément va dans l'autre sens *e.g.* la porte se referme. Nous supposons que le moteur réalise trois actions :

2. <http://www.lejos.org/ev3/docs/lejos/hardware/sensor/EV3ColorSensor.html>

pousser, tirer et arrêter. Ces actions sont exécutées lorsque le contrôleur lui demande de le faire. Son comportement est alors celui de la figure 11.

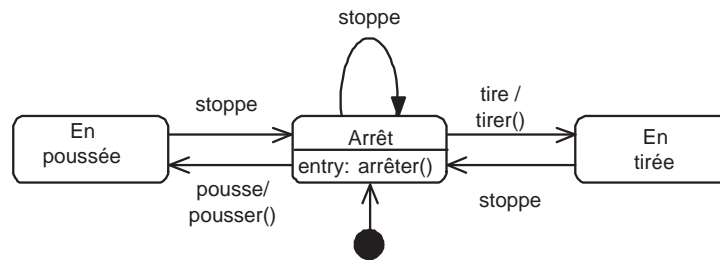


Figure 11 : Diagramme Etats-transitions du moteur simple

Pour l'EV3, un moteur court est de type translation et un moteur long est de de type rotation.

3.4.2 Elément actionné (bras, porte, chariot...)

La modélisation du comportement de l'élément actionné par un moteur se fait en fonction du contexte (le côté fonctionnel de l'application). La variable `mt` correspond au moteur qui commande la portail. Par exemple, le comportement générique d'une porte peut être décrit par la figure 12.

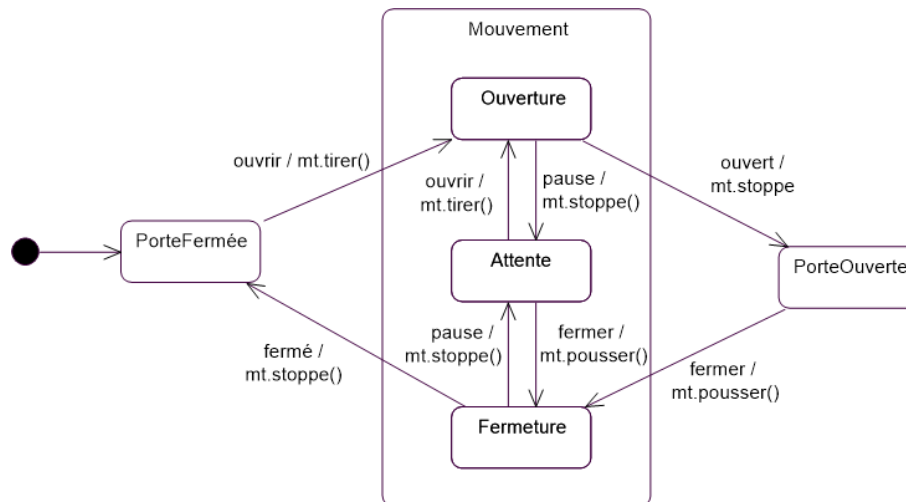


Figure 12 : Diagramme Etats-transitions de la porte - portail

Dans le détail, la porte peut être dans un des trois états suivants : fermée, ouverte ou en mouvement. Elle se bloque dès que le contrôleur le décide. Cet arrêt dure jusqu'à ce qu'une seconde pression sur la télécommande soit détectée, ce qui aura pour effet de relancer le mouvement dans le sens demandé. La portail peut donc être dans cinq états distincts, `porteOuvverte`, `porteFermée`, `Attente`, `Ouvverture` et `Fermeture`. L'état `Mouvement` est en fait un super-état (il regroupe `Ouvverture`, `Attente` et `Fermeture`) et le diagramme final est un automate hiérarchique.

Au niveau spécifique du dispositif actionné, on pourra ajouter des temporisations, des caractéristiques techniques ou des états intermédiaires.

3.5 Contrôleur

Le contrôleur reçoit les signaux des capteurs et de l'interface de pilotage et commande le matériel asservi. Son rôle est d'analyser ces signaux et de décider, en fonction de ceux-ci, ce que doit faire le portail. Pour décrire nous pourrions nous baser sur un produit des automates des comportement des battants mais nous avons choisi une description plus spécifique. Les états du contrôleur correspondent, en fait, à la coordination des deux battants qu'il faut abstraire au niveau du portail, aux différentes étapes dans son fonctionnement.

L'analyse devra fournir le comportement du contrôleur (diagramme état-transition) spécifique au poste de travail traité. Nous ne pouvons en dire plus car cela dépend du poste de travail cible. On trouvera des exemples dans les cas d'étude des années passées (porte de garage, portail, contrôle de lumière, véhicule...).

La conception générale en donnera une version instrumentalisée par exemple sous forme de *pattern*

4 Système logiciel global

Le système logiciel est centralisé autour du contrôleur. Nous en donnons un modèle générique sous forme du diagramme de classes de la figure ??

La structure matérielle est constituée d'un contrôleur, de différents composants matériels asservis équipés chacun d'un moteur, d'une éventuelle lumière clignotante, d'une interface de pilotage, d'un capteur de présence et/ou de couleur ainsi que de capteurs de contact. Dans une version abstraite du diagramme de classes, , Nous n'avons aucune information nous permettant de décrire les attributs de l'un quelconque de ces constituants. Par contre, nous avons uniquement des opérations génériques qu'il faudra affiner pour chaque poste de travail (ou ressource).

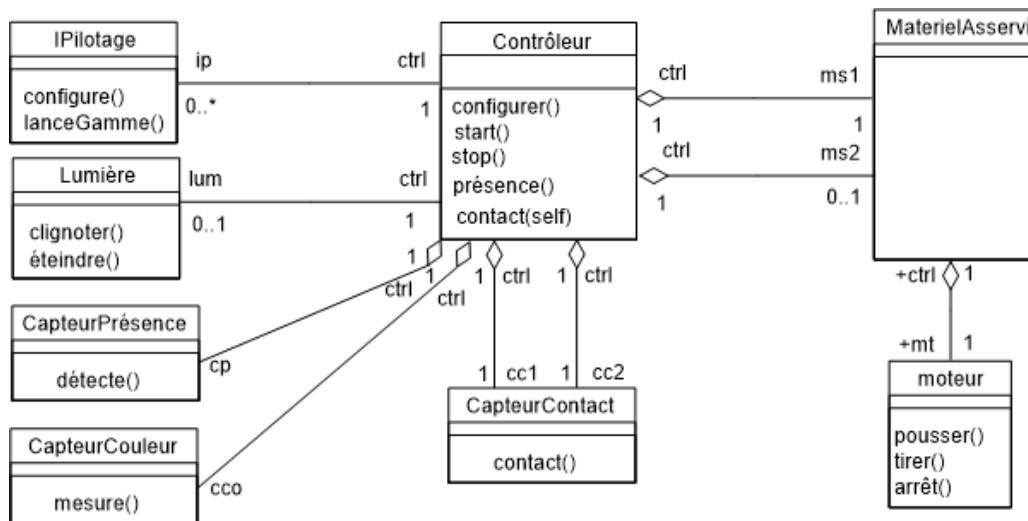


Figure 13 : Diagramme de classes - modèle logique générique - Cas workshop

Nous pourrions, par la suite (durant la phase de conception) ajouter éventuellement un attribut état dans chacune des classes ayant un comportement dynamique (i.e. représenté par un diagramme états-transitions), cet attribut servant de support d'enregistrement de l'état courant. Cela risque toutefois de faire double emploi avec un attribut du métamodèle.

4.1 Déploiement et conception

En pratique, l'interface de pilotage sera déployée sur un dispositif mobile qui communiquera avec le contrôleur qui sera lui déployé sur l'EV3. Les capteurs et actionneurs sont reliés par fil avec une API intégrée selon le support technique choisi dans l'analyse technique.

4.2 Protocoles de communication

Les dispositifs physique échangent directement avec le contrôleur. On suppose donc un système simple d'échanges de messages en circuit fermé.

La télécommande communique à distance avec le contrôleur pour commander l'ouverture et la fermeture des portails. En pratique, la communication avec LEGO MINDSTORMS EV3 se fait en bluetooth ou Wifi. En extension, un protocole sécurisé vérifie l'authenticité de la télécommande pour ces échanges.

5 Exigences, contraintes et hypothèses

Les exigences fonctionnelles sont fournies dans un document joint.