Domodoor

Dossier de specifications formelles

Laureen LOISON-Maxime VIREY-Antoine GUERON
UNIVERSITE DE NANTES – MIAGE ALT Module de development logiciel

Sommaire

Introduction

Le module de développement logiciel : Approche Formelle & à Objets fait suite au module de Génie Logiciel à Objet. L'objectif de ce module est de nous sensibiliser à l'approche formelle lorsque nous développons un projet. Dans cet objectif, il est nécessaire d'aborder la spécification du logiciel souhaité. Après les spécifications, le raffinage et les preuves des propriétés, il sera plus aisé de développer le code du logiciel final.
Le dossier reprend la spécification détaillée du projet DomoDoor pour savoir comment interagissent les composants entre eux. Chaque composant possède ses propres actions, spécificités et relations qui méritent d'être explicitées, ainsi que nos choix techniques et fonctionnels.
Le temps à consacrer à ce projet est conséquent, induit principalement par le manque de documentation et de communauté sur la notation Z. Nous passons donc plus de temps à essayer de comprendre la notation, ce qui nous ralenti.
Nous allons dans un premier temps présenter nos démarches et la méthodologie adoptée pour la mise en œuvre. Ensuite, nous présenterons nos hypothèses et nos choix à partir desquels nous avons pensé construire notre système. Enfin, nous présenterons les résultats des spécifications.

I- La mise en œuvre et la méthodologie

Dans cette section, nous allons parler de la méthodologie adoptée pour la rédaction des spécifications.

A- La mise en œuvre

Étant un groupe de 3 personnes, et pour être le plus efficace possible, nous avons décidé de nous répartir les tâches de travail. Une personne qui se consacre à la rédaction du rapport et les deux autres à la transformation du diagramme de classe fournit en notation Z. Nous avons utilisé la méthode du pair programming lorsque nous en avions l'occasion afin de pouvoir s'entraider sur la conversion du modèle E-A-P (Entités-Associations-Propriétés) vers la notation Z.

B- La méthodologie

Pour réaliser la conversion du diagramme de classe en notation Z, nous avons utilisé la méthode vue en cours. Elle consiste à passer le diagramme de classe en modèle E-A-P. Puis de passer ce dernier en notation Z grâce à plusieurs formalismes.

Convertir un diagramme de classe en modèle E-A-P

Nous allons partir du diagramme de classe fournit dans le document de l'étude de cas. Le voici ci-dessous pour rappel :

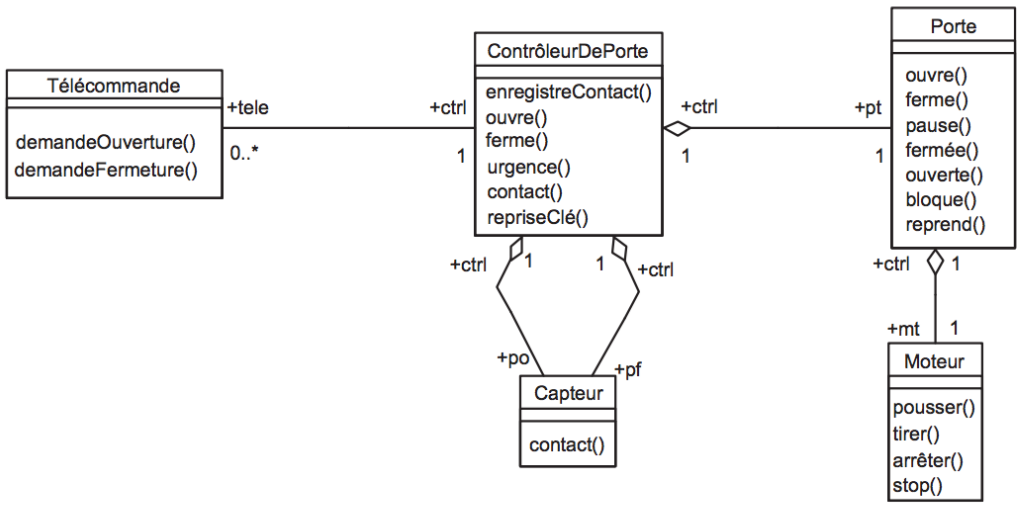


Figure 7 : *Diagramme de classes - Porte*

Pour passer du diagramme ci-dessus au modèle E-A-P, il faut convertir les classes en entité, adapter les associations ainsi que les cardinalités.
Nous distinguons ici, les entités suivantes :

- Telecommande
- Controleur
- Porte

- Moteur
- Capteur

Ensuite, pour les associations, on retrouve les suivantes :

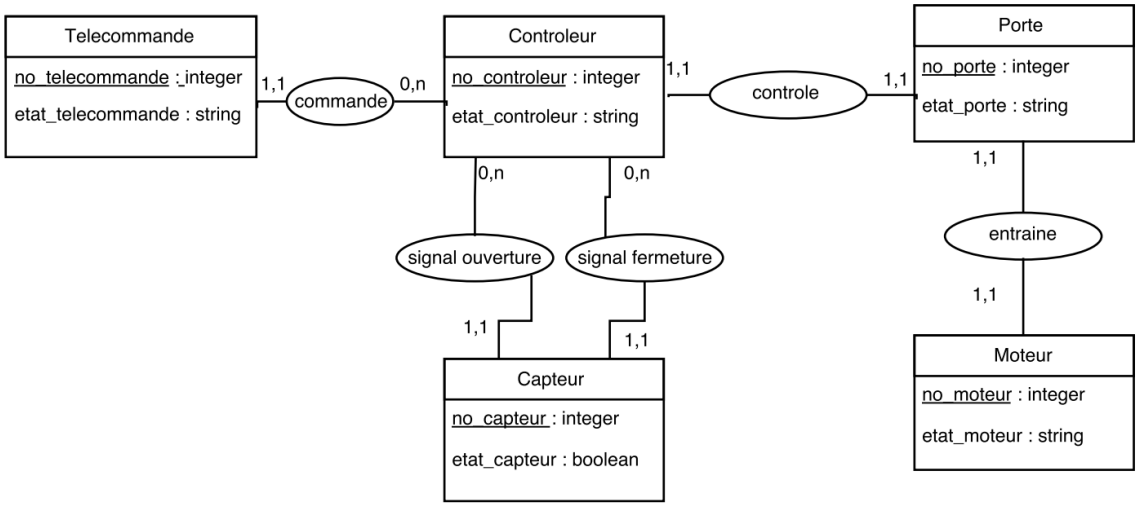
- Commande
- Contrôle
- Signal ouverture
- Signal fermeture
- Entraine

Enfin, pour les cardinalités présentes sur le diagramme de classe (0,1,N), il faudra les convertir dans la notation merise. Elles sont inversées dans les 2 formalismes. Par exemple, 1 devient 1,1 ou 0..n devient 0,n.

Une fois ces étapes terminées, on peut commencer à compléter les entités du modèle E-A-P avec des attributs.

Après conversion de ce diagramme de classe, nous obtenons le modèle E-A-P ci-dessous :

Modèle E-A-P



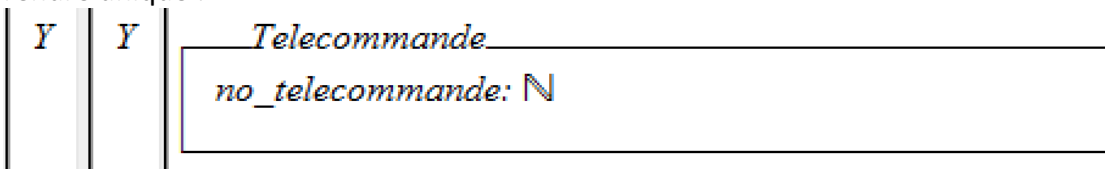
Convertir le modèle E-A-P en notation Z

D'après le modèle ci-dessus, nous avons pu distinguer les schémas à déclarer en Z. Tout d'abord, nous avons procédé à l'écriture des entités en Z. Par exemple, l'entité *Telecommande* en notation Z va se représenter de la manière suivante :
On déclare une énumération qui représentera les états qu'une télécommande peut prendre :

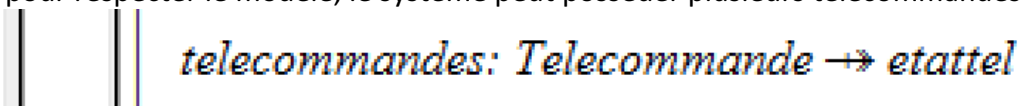
$$Y \mid Y \mid etattel ::= actif$$

Pour le moment, nous savons qu'une télécommande n'a qu'un seul état possible. Mais nous avons fait le choix de créer une énumération au cas où à l'avenir on souhaite rajouter des états que pourraient prendre la télécommande.

Puis, on va attribuer un numéro à la télécommande (id), ce qui permettra également de la rendre unique :



Enfin, pour respecter le modèle, le système peut posséder plusieurs télécommandes :



Ci-dessus, nous déclarons un ensemble *telecommandes*, qui associe à une télécommande un état.

Nous avons procédé de la même façon pour les entités *Controleur*, *Porte*, *Moteur* et *Capteur*.

II- Les hypothèses et choix

Dans cette section, nous allons parler des différents choix fonctionnels que nous avons choisi d'implémenter.

A- Choix d'implémentation des capteurs

Le portail sera muni de 2 capteurs, l'un en bas et l'autre en haut de la porte. Chaque capteur va signaler le contact de la porte par la fonction *enregistrerContact* du contrôleur.

B- Détermination de l'état de la porte

Pour savoir dans quel état se trouve la porte, nous allons nous baser sur l'état des capteurs et l'état du moteur. Nous pouvons distinguer plusieurs cas :

- Dans le premier, le capteur du haut avait enregistré un contact et l'état du moteur est 'enPoussee', dans ce cas l'état de la porte est 'enFermeture'.
- Dans le second cas, le capteur du bas avait enregistré un contact et l'état du moteur est 'enTiree', dans ce cas l'état de la porte est 'enOuverture'.
- Dans un troisième cas, l'état du moteur est 'arret' après avoir été en activité, et aucun capteur n'a signalé de contact. Dans ce cas l'état de la porte est bloquée dans l'état 'porteArretee'.

On pourrait également se baser sur l'état du contrôleur pour déterminer l'état de la porte. Car si le moteur est dans l'état 'ouverture', alors la porte est 'enOuverture', lorsque le contrôleur est dans l'état 'fermeture', alors la porte est 'enFermeture'.
La porte serait dans l'état 'porteArretee' lorsque le contrôleur serait dans l'état 'enAttente'.
Enfin, l'état 'urgence' du contrôleur ferait passer la porte dans l'état 'porteBloquee'.

Lorsque le bouton d'urgence sera pressé, l'état de la porte sera dans l'état 'porteBloquee' jusqu'à ce que la situation se débloque par l'insertion de la clé.

C- Détermination de l'état du moteur

Lorsque l'on appuie sur le bouton de la télécommande, le contrôleur va agir sur le moteur de la porte. Plusieurs cas sont à différencier :

- Si la porte est dans l'état 'porteFermee', alors le moteur va passer dans l'état 'enTiree'.
- Si la porte est dans l'état 'porteOuverte', alors le moteur va passer dans l'état 'enPoussee'.
- Si la porte est dans l'état 'enOuverture' ou 'enFermeture' alors le moteur va passer dans l'état 'arret' et la porte passera dans l'état 'porteArretee'.
- Si la porte est dans l'état 'porteArretee', alors le moteur va passer dans l'état dans lequel il était avant d'être dans l'état 'arret'.

Enfin, si la porte est dans l'état 'porteBloquee', alors seule l'activation avec la clé permettra à la porte de passer dans l'état 'porteArretee'.

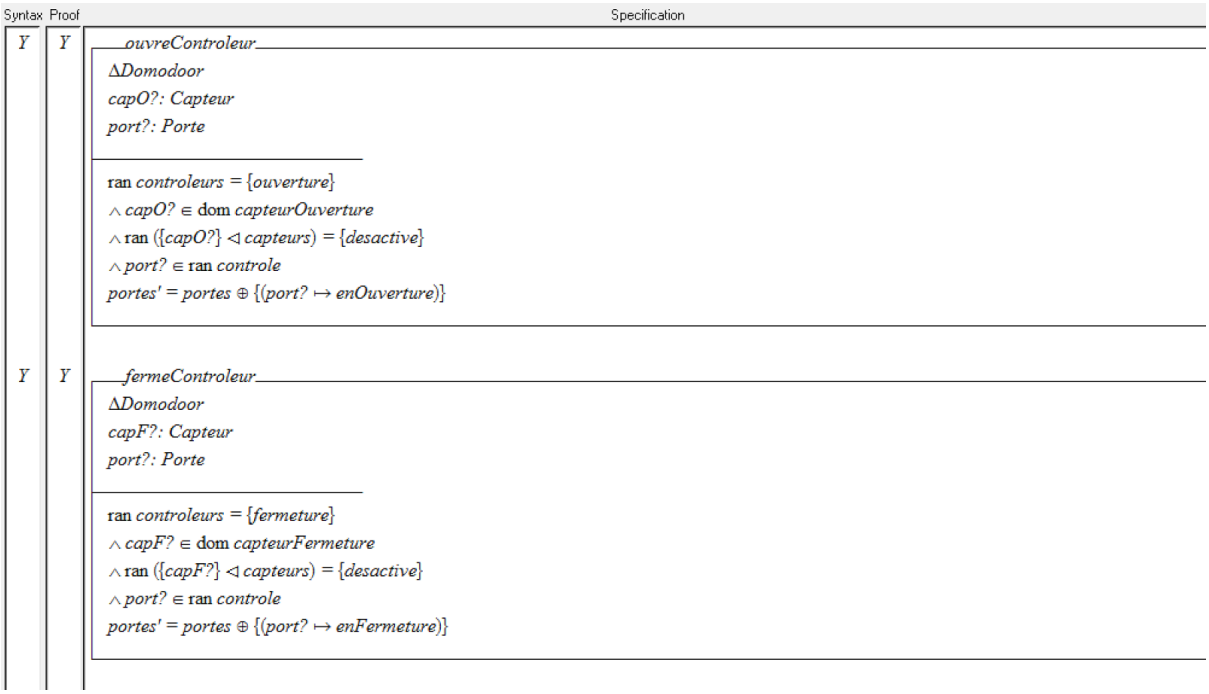
D- Détermination de l'état du contrôleur

L'état du contrôleur va être déterminé après analyse de l'état des autres entités.
Si le bouton d'urgence a été activé, alors le contrôleur va passer dans l'état 'urgence'.
Lorsque le capteur du bas de la porte enregistre un contact, le contrôleur passe dans l'état 'enAttente'.
Lorsque le capteur du haut de la porte enregistre un contact, le contrôleur passe dans l'état 'enAttente'.
Lorsque le contrôleur est dans l'état 'enAttente', que la porte est dans l'état 'porteFermee' et que la télécommande envoie la demande d'ouverture, alors le contrôleur passe dans l'état 'ouverture'.
Lorsque le contrôleur est dans l'état 'enAttente', que la porte est dans l'état 'porteOuverte' et que la télécommande envoie la demande de fermeture, alors le contrôleur passe dans l'état 'fermeture'.

E- Détermination de l'état de la télécommande

L'état de la télécommande est toujours actif. Il n'existe pas dans cette version du programme un autre état possible pour la télécommande.

III- Les résultats



Sur la capture d'écran ci-dessus, nous pouvons voir les modélisations en notation Z des fonctions ouvre() et ferme() du contrôleurs.
Explicitions la notation de ouvreControleur :
La spécification Z ouvreControleur modifie l'état du système Domodoor (représenté par le delta).
Elle prend en entrée 2 paramètres captO (le capteur d'ouverture) et port (la porte). Les paramètres d'entrées sont représentés par un nom suivi d'un '?'.
Ensuite, dans la partie des prédicats, on précise certaines propriétés :

- Le contrôleur est dans l'état 'ouverture',
- Les capteurs d'ouverture appartiennent aux capteurs,
- La porte passe dans l'état 'enOuverture',
- La porte fait partie des portes.

Nous avons ainsi modélisé chacune des fonctions qui sont présentes dans le diagramme de classe en notation Z. Le but étant de prouver fonctionnellement les fonctions.

Conclusion

Ce projet nous a permis de prendre conscience de l'importance de l'utilisation des méthodes formelles dans un projet. En revanche, on comprend aussi pour quelles raisons très peu d'entreprises passent par cette étape, car cela demande beaucoup de temps.
Nous avons tout de même apprécié de pouvoir constater nos erreurs de conception avant d'avoir commencé les développements pour éviter de perdre du temps à faire et défaire du code.
Nous avons pu appliquer certains concepts des méthodes agiles (comme le pair programming), et d'autres concepts vus préalablement en cours afin d'en constater l'utilité dans un cas plus concret.