

Modélisation d'une barrière de péage



1. Cahier des charges

L'application a pour but d'informatiser une barrière de péage d'autoroute pleine voie ne comportant qu'un seul sens de circulation.

Une barrière de péage est constituée de différentes bornes de péage et d'un poste de supervision qui gère le fonctionnement de l'ensemble des bornes. Nous appellerons borne de péage une voie de circulation équipée pour la perception du péage. Elle est ainsi constituée d'un bandeau en amont du péage qui indique l'état de la voie, ouverte ou fermée, une barrière manuelle en amont, deux boucles magnétiques : une avant la barrière aval et une après, qui détectent le type de véhicule ainsi que son passage, un dispositif de paiement et enfin une barrière aval couplée à un feu de signalisation. La Figure 1 résume l'organisation structurelle d'une borne de péage.

1.1 Différents types de borne

1.1.1 Borne manuelle

Il s'agit d'un guichet avec un opérateur humain qui encaisse le péage. Tous les types de véhicules sont acceptés.

Le guichetier dispose d'une caisse, d'un lecteur de carte et de trois boutons :

- Un bouton pour gérer l'ouverture de la barrière aval et le feu associé.
- Un bouton alarme en cas de problème.
- Un bouton permettant la gestion du bandeau amont, ouvert ou fermé.

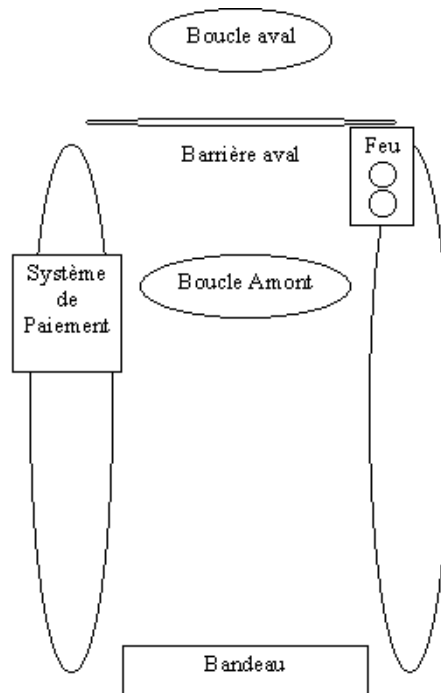


Figure 1: Schéma d'une borne de péage.

1.1.2 Borne de télépéage

Les bornes de télépéage sont équipées d'un gabarit permettant uniquement aux voitures de passer. L'usager dispose d'un bouton d'appel en cas de problème et un détecteur identifie son badge pour effectuer le paiement, ouvrir la barrière aval et changer la couleur du feu.

1.1.3 Borne automatique

Les bornes automatiques sont, elles aussi, réservées aux voitures et disposent également d'un gabarit. L'usager dispose de même d'un bouton d'appel en cas de problème. Elles acceptent les paiements par carte bleue, carte d'abonnement et en monnaie. La borne détecte les fausses pièces et rend la monnaie.

1.2 Description fonctionnelle

1.2.1 Borne manuelle

Tout d'abord, le guichetier détermine le type de véhicule. Ensuite il fait payer soit par monnaie, soit par carte bancaire ou carte d'abonnement. Dans le cas d'un paiement par carte d'abonnement la caisse effectue une demande d'accord au site central avec débit immédiat. Enfin il ouvre la barrière amont puis met le feu au vert. Le véhicule part, lors de son passage sur la boucle aval le feu passe au rouge, la barrière aval se ferme et un rapport est envoyé au poste de supervision.

La boucle au sol amont permet de vérifier la concordance du type de véhicule avec les données saisies par le guichetier. La borne enregistre aussi les paiements par carte bancaire pour traitement ultérieur. Pratiquement, elle envoie les données au superviseur qui les stocke jusqu'à la fin de la journée.

1.2.2 Borne de télépéage

Lors de la détection d'un badge, la borne demande un accord de paiement au site central, lorsque ce paiement est accepté, le compte est débité, le feu passe au vert, la barrière se lève. Après le passage du véhicule sur la boucle aval, la borne met le feu au rouge, ferme la barrière et envoie un rapport au poste de supervision.

Le site central gère un temps minimum entre deux passages avec le même badge.

1.2.3 Borne automatique

Lors de la détection d'un véhicule par la boucle amont, la borne fait payer soit par monnaie, soit par carte bancaire ou carte d'abonnement, pour un paiement par carte d'abonnement comme pour les autres bornes une demande est faite au site central. Les informations liées aux cartes bancaires sont envoyées au poste de supervision.

Lorsque ce paiement est effectué, le feu passe au vert, la barrière se lève. Après le passage du véhicule sur la boucle aval, la borne met le feu au rouge, ferme la barrière et envoie un rapport au poste de supervision.

Le manque de monnaie dans le monnayeur de la borne lève une alarme. Le rendu de monnaie est alors approximatif par excès. S'il se prolonge, il entraîne la fermeture de la voie, par la borne elle-même.

1.3 Autres caractéristiques de fonctionnement

Les véhicules spéciaux, comme les voitures de police ou de pompiers, disposent d'un passage gratuit enregistré par le guichetier ou les bornes automatiques, grâce à un badge ou une carte "gratuit". Nous prenons ici seulement en compte leur passage à des fins statistiques.

Lorsque la boucle au sol aval signale la sortie du véhicule, nous refermons la barrière et nous remettons le feu au rouge. Le fait de passer la barrière aval entraîne l'émission d'un rapport comprenant le type de véhicule, le numéro de voie, l'heure et la somme perçue.

Le superviseur constitue le système de gestion de la barrière. Il existe un superviseur par barrière de péage. C'est une entité externe à notre système à qui nous envoyons des rapports.

Un message est envoyé au superviseur lorsque le trafic est important et qu'il est possible d'ouvrir de nouvelles voies. Ou inversement lorsque le trafic est trop faible pour le nombre de voies ouvertes. Le système vérifie aussi qu'il y a au moins une voie

ouverte.

Le superviseur :

- Il est le seul à décider de l'ouverture d'une voie.
- Il reçoit les alarmes et les traite.

Le système de gestion :

- Il comptabilise les flux financiers.
- Il enregistre les rapports et les alarmes.
- Il effectue des statistiques en fonctions des rapports.

Le site central est donné ici à titre informatif :

- Il vérifie le temps minimum entre deux passages d'un badge de télépéage.
- Il effectue en fin de journée les transactions pour les cartes bancaires, les données proviennent des superviseurs des différentes barrières d'autoroute.
- Il débite immédiatement les comptes pour les cartes d'abonnement et les badges de télépéage.

1.4 Les alarmes

Voici les différentes alarmes possibles reçues par le poste de supervision :

- L'utilisateur ou le guichetier appuie sur le bouton alarme.
- La borne signale que sa barrière aval ne se lève pas.
- La voie est fermée et la boucle au sol aval détecte un véhicule, ou le système de paiement reçoit des pièces. (Ceci ne sera pas implémenté !)
- Plus d'un véhicule est passé, détection par la boucle aval, une sonnerie sera également déclenchée.
- Le refus du site central pour un paiement par abonnement à la demande de la borne.
- Manque de monnaie pour les bornes automatiques.

Les alarmes entraînent l'émission d'un rapport.

2. Travail à réaliser

2.1 Objectif

Vous devrez réaliser un logiciel qui simule le comportement d'une barrière de péage.

Ce logiciel est une version complètement virtuelle du système réel. Ce qui est attendu, c'est un logiciel fonctionnel qui permet au client de pouvoir dimensionner son système réel avant sa réalisation effective.

L'enseignant joue le rôle de votre client. Vous pouvez lui faire toute suggestion d'amélioration du produit. Mais, c'est toujours lui qui décide de l'opportunité d'ajouter cette fonctionnalité au produit final.

2.2 Les contraintes sur la gestion de projet

La réalisation et l'organisation du travail sont soumises à un certain nombre de contraintes :

- Il vous est demandé d'utiliser une **méthode de développement structurée** pour laquelle le code est la résultante des phases d'analyse (voir la section 3 suivante ainsi que le chapitre 3 du cours de génie logiciel 1^{re} année sur la plateforme pédagogique).
- La rédaction des diagrammes UML devra se faire avec un atelier de génie logiciel tel que **ArgoUML** ou un autre atelier de votre choix.
- Ce travail intègre la définition et le codage des tests unitaires qui permettront d'évaluer le logiciel à produire. Vous devez utiliser **JUnit** pour la définition des tests (voir la documentation dans le cours de génie logiciel 1^{re} année sur la plateforme pédagogique).
- Le travail devra être réalisé en **Java**.
- L'architecture devra bien entendu faire appel aux **Design Patterns**, dont le choix devra être pertinent et justifié.
- Vous devez absolument utiliser **git** pour l'archivage et le partage des différents fichiers produits. La récupération du code pour la notation sera faite par l'enseignant directement sur le dépôt git.

3. Développement logiciel basé sur une méthode structurée de type "processus unifié"

3.1 Phases de développement d'un projet

Le développement d'un projet informatique distingue 4 phases de maturité du logiciel :

1. *Étude d'opportunité*. Cette phase d'avant-projet doit conduire à la rédaction d'un cahier des charges définissant les besoins du client en détaillant les fonctionnalités attendues du futur système, les contraintes associées et une analyse des risques de ne pas livrer le logiciel attendu. Parallèlement au cahier

des charges, on doit aussi disposer d'une estimation de la planification envisagée pour les phases suivantes. Cela correspond ici aux 2 premières semaines.

2. *Élaboration*. Elle se focalise sur la définition de l'architecture la plus adaptée et la production d'un premier prototype de démonstration et validation du concept. . Cette phase devra être terminée pour la semaine 5
3. *Construction*. C'est ici qu'est développé puis intégré concrètement le logiciel et que sont appliqués les tests unitaires et fonctionnels.
4. *Transition*. C'est la phase terminale qui a pour but d'installer le système développé dans son environnement de fonctionnement définitif et d'appliquer les tests de qualification et de recette. Cela correspond à la fin du projet, soit la semaine 8.

3.2 Cycle itératif

Chaque phase précédente procède d'un ou de plusieurs cycles itératifs et se conclut par la livraison d'un prototype (ou d'une version du logiciel pour la dernière itération). Chaque itération suit un cycle de développement en V composé des quatre étapes :

1. Définition et analyse des besoins d'une itération
2. Spécifications fonctionnelles (avec la langage UML)
3. Conception / Codage
4. Intégration, Tests

Une itération correspond à la sélection d'un ensemble de tâches à réaliser. Le choix des tâches doit être **guidée par l'étude des risques**. On privilégie les tâches qui correspondent aux risques les plus forts.

La durée d'une itération doit être courte. En effet, pour éviter l'effet big-bang, l'intégration doit être faite de façon continue. Dans notre contexte, elle doit être de l'ordre de la séance de TP. Cela signifie que l'intégration doit être réalisé au moins une fois dans chaque séance de TP et cela s'accompagne du lancement des tests déjà programmés pour vérifier qu'il n'y a pas de régression.

3.3 Tests et validation

Le **cycle V** nécessite la définition préalable et la conduite de tests en regard de chaque étape :

- | | | |
|---------------------------------------|---|--|
| - Analyse des besoins d'une itération | → | Qualification système, test de recette |
| - Spécifications fonctionnelles | → | Validation système |
| - Conception | → | Tests d'intégration |

– Codage

→ Tests unitaires et qualification de chaque composant.

Les tests unitaires et les tests d'intégration doivent être définis lors de la phase d'analyse sur papier et programmés avec JUnit (voir la documentation de JUnit sur la plate-forme pédagogique).

4. Les échéances et les livrables

4.1 Livrable à S+2 : le cahier des charges

Le cahier des charges comportera au moins :

- Un diagramme des cas d'utilisation spécifiant les fonctionnalités du logiciel à développer et toute documentation textuelle utile détaillant les cas d'utilisation complexes.
- Une analyse des risques de ne pas livrer le logiciel demandé et un plan de gestion des risques et des points critiques.
- Une architecture préliminaire du système avec identification des constituants principaux à développer ou à réutiliser.
- Un diagramme de Gantt prévisionnel avec la durée des étapes prévisionnelles du projet, qui sera actualisé chaque semaine.

4.2 Livrables à S+5 : un premier prototype fonctionnel

Ce premier prototype permettra de valider l'architecture du système et quelques-unes des fonctionnalités, notamment les plus risquées.

La présentation se fera **sous la forme d'une démonstration courte**. Cette démonstration devra être préparée pour être performante et efficace (le fameux « effet démo » ne sera pas considéré comme une excuse, mais comme une faute).

4.3 Livrable à S+8 : le logiciel final

Le compte rendu final de TP devra comporter tout document que vous jugerez utile pour présenter la modélisation ainsi que l'organisation de votre projet. Il devra par contre obligatoirement comporter :

- Les diagrammes UML représentatifs du modèle.
- Une liste d'exigences relatives à l'ergonomie, l'IHM, la visualisation, etc.
- Une documentation détaillant le rôle de chacun des membres du groupe à chacune des étapes ainsi que le temps consacré à leur réalisation (arborescence de configuration, planification et organigramme des tâches, suivi, ...).

- La description et compte-rendu des tests en regard de chaque étape du cycle en V utilisé, ainsi que les moyens d'essai associés.

Le travail fera l'objet d'une présentation orale.

5. Évaluation du travail

Compte-tenu du temps imparti pour le projet, ce qui sera évalué c'est d'abord la façon dont le projet a été organisé (division-intégration), la pertinence de la modélisation proposée et la définition des tests proposés en regard des étapes de développement. Il ne sera ainsi exigé qu'un prototype du produit correspondant à la modélisation proposée.

| Points évalués |
|--|
| Livrable S+2 |
| Document de description de l'organisation de projet <ul style="list-style-type: none"> - Diagramme de Gantt - Division des tâches - Analyse des risques Document de définition des besoins <ul style="list-style-type: none"> - Cas d'utilisation - Maquette (papier suffit) - Diagramme des classes du domaine - Définition des tests (sur papier) |
| Livrable S+5 |
| Démonstration d'un prototype |
| Livrable S+8 |
| Exposé du résultat d'un travail accompli Démonstration du produit Dossier et rapport |