

Cours Modélisation et Validation de logiciels

Fiche 6 : Modélisation et Validation d'un Ascenseur

Philippe Dhaussy,
Lab-STICC, ENSTA Bretagne
philippe.dhaussy@ensta-bretagne.fr

Novembre 2017

1. Modélisation d'un ascenseur à N étages

On veut modéliser le fonctionnement d'un ascenseur à N étages. Le modèle de l'ascenseur contient 4 type d'automates : **Cage**, **Bouton**, **Porte** et **Client** (figure 1). Les instances de **Bouton**, **Porte** et **Client** correspondent respectivement aux boutons, aux portes et aux clients à chaque étage. La synchronisation entre les processus s'effectuent par des canaux.

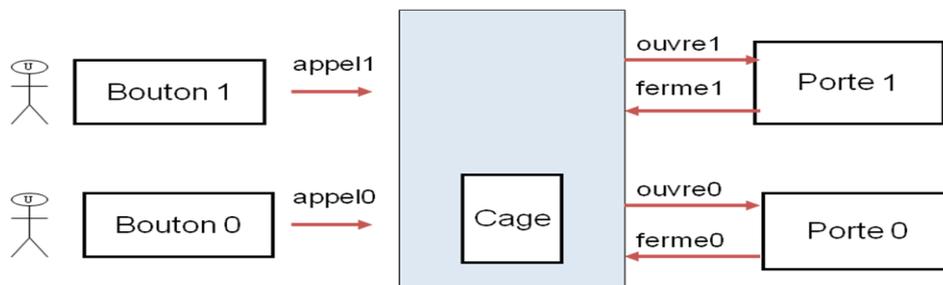


Figure 1 : modèle d'un ascenseur à n étages.

A chaque étage, un client arrive et se comporte de la manière suivante :

- Le client se présente devant l'ascenseur. Il ne peut appuyer sur le bouton que quand la porte de l'ascenseur est fermée. Si la porte est ouverte, c'est que la cage est déjà au bon étage. Donc selon le cas, il appelle (si porte fermée) ou non (si porte ouverte) l'ascenseur.
- Après avoir appuyé sur le bouton d'appel, il attend l'arrivée de l'ascenseur (optionnel car la cage est peut-être déjà à l'étage). Ensuite il attend que la porte s'ouvre et entre dans la cage.
- Une fois dans la cage, il attend la fermeture de la porte et ensuite il appui sur le boîtier interne à la cage pour choisir un étage (optionnel car il peut ressortir aussitôt). On considérera un choix aléatoire de l'étage. La porte doit être fermée avant la prise en compte de l'étage demandée.
- Il attend la fin de déplacement de l'ascenseur (optionnel il peut avoir choisi l'étage auquel il se trouve déjà). Une fois arrivée à l'étage demandé, il attend l'ouverture de porte et sort de l'ascenseur.
- Par simplification, on considère que, si la cage est en cours de déplacement pour servir un client, aucune demande d'un autre client ne peut être prise en compte jusqu'à ce que le client précédent soit sorti de l'ascenseur (pas de mémorisation cumulée des demandes).

Le comportement de la **cage** de l'ascenseur se comporte de la manière suivante :

- Si la cage est déjà à l'étage demandé ou si la porte est ouverte, la cage ne bouge pas.
- Si la cage est au-dessus de l'étage demandé et si la porte de l'étage courant est fermée, la cage descend jusqu'à l'étage (la cage passe par un état intermédiaire *descend*).
- Si la cage est en dessous de l'étage demandé et si la porte de l'étage courant est fermée, la cage monte jusqu'à l'étage (la cage passe par un état intermédiaire *monte*).

Le comportement d'une **porte** d'étage se comporte de la manière suivante :

- Si la cage est à un étage, la porte de cet étage peut s'ouvrir (elle est débloquée).
- Sinon, la porte est fermée et ne peut s'ouvrir (elle est bloquée).

Le modèle est temporisé pour indiquer des temps de montée et de descente de la cage de l'ascenseur. Par exemple, on considère que pour passer d'un état n à $n+1$ ou à $n-1$, la durée est de 1 unité de temps. Pour les clients, on pourra considérer qu'il n'attendent pas indéfiniment hors de l'ascenseur.

Travail à réaliser

- Modéliser l'application sous la forme d'un réseau d'automates UPPAAL.
- Simuler.
- Vérifier les propriétés suivantes :
 - Le modèle n'est jamais en deadlock
 - Si la cage est à l'étage n , toutes les portes des étages différents de n sont fermées.
 - Si la cage est à l'étage n et que la demande est faite pour la conduire à l'étage n' , la cage se trouvera fatalement à l'étage n' (n et n' sont compris entre 0 et $N-1$).
 - Si un client, à l'étage n demande à aller à l'étage n' , il finira par sortir de l'ascenseur à l'étage n' .
 - Si un client rentre dans l'ascenseur, il sera à l'étage demandé avant un certain délai (le délai dépendra des valeurs temporelles que vous avez choisies).