

## DIAGRAMME D'ÉTATS – ASCENSEUR



Le Musée des Arts Contemporains de Lyon (le MAC Lyon) est équipé d'un ascenseur à l'usage des visiteurs afin qu'ils accèdent aux différents étages de la collection temporaire en limitant le temps d'attente sur le palier de chaque niveau.

Il est composé de **2 cabines** desservant les étages, 3 niveaux de collection et le rez-de-chaussée. Les usagers doivent appeler une des cabines pour **monter** ou **descendre** et le système doit **réagir**, en fonction des appels et du déplacement en cours, le plus intelligemment possible pour répondre aux demandes et

faire gagner du temps à l'ensemble des usagers.

### 1 OBJECTIFS

*Les objectifs de l'étude de l'ascenseur sont la modélisation de la commande d'un Système à Evénements Discrets (SED) en utilisant les outils SysML (diagramme de séquence et diagramme d'états) et les algorithmes.*

Vous avez à votre disposition les modèles des ascenseurs simple et double sous matlab (matlab/simulink/stateflow). Dans la fenêtre simulink, vous avez la partie commande et la partie mécanique de l'ascenseur et **le diagramme d'état à modifier**. En annexe, vous trouverez un tutoriel pour l'utilisation de stateflow.

Un document sur la construction des algorithmes est fourni en annexe 2.

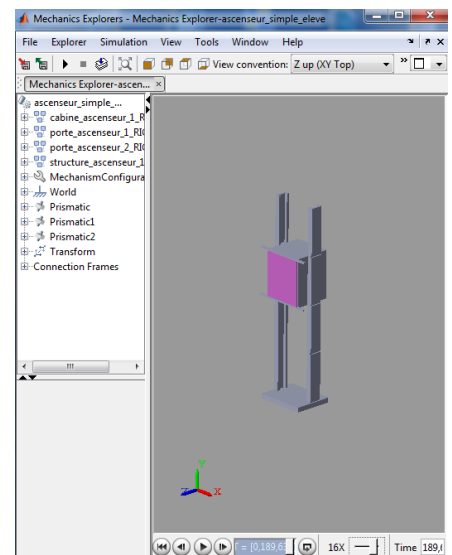
Le travail demandé est réalisé par des équipes de 3 ou 4 étudiants. Pour chaque partie, vous rédigez quelques lignes d'explications et vous serez particulièrement attentifs à la clarté de vos schémas et diagrammes.

### 2 LIMITES DE L'ETUDE DE L'ASCENSEUR

Trois acteurs ont été identifiés (rappelons qu'un acteur est un élément extérieur au système et qui interagit avec lui) :

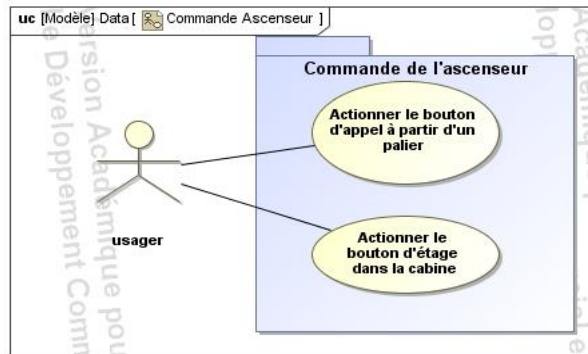
- L'utilisateur : personne désirant se rendre à n'importe quel étage du musée
- L'administrateur : personne pouvant accéder au système et agir sur son fonctionnement
- L'agent de maintenance : personne appelée à chaque problème ou à chaque révision

Nous limiterons notre étude au cas d'utilisation de l'utilisateur et pour une commande en fonctionnement normal (pas de panne ni maintenance).



### 3 PARTIE I : PRESENTATION SysML DU FONCTIONNEMENT DE L'ASCENSEUR

#### 3.1 DIAGRAMME DE CAS D'UTILISATION



#### 3.2 LES BLOCKS PARTICIPANTS A L'EXECUTION DU CAS D'UTILISATION

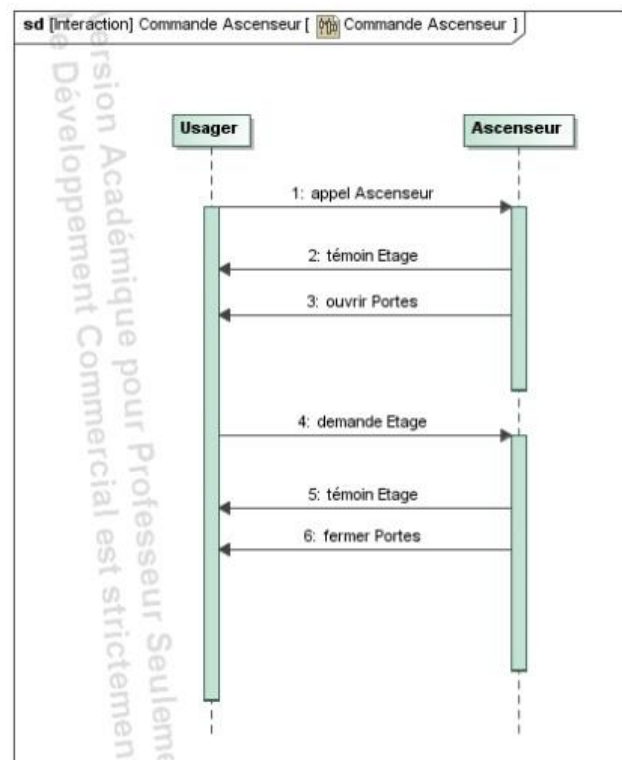
- Usager
- boutonEtage (dans la cabine)
- boutonAscenseur (bouton qui permet l'appel de l'ascenseur)
- controleurAscenseur
- Ascenseur
- Portes

#### 3.3 SCENARIO DE BASE

Le scénario de base pouvant être extrait du cas d'utilisation est le suivant :

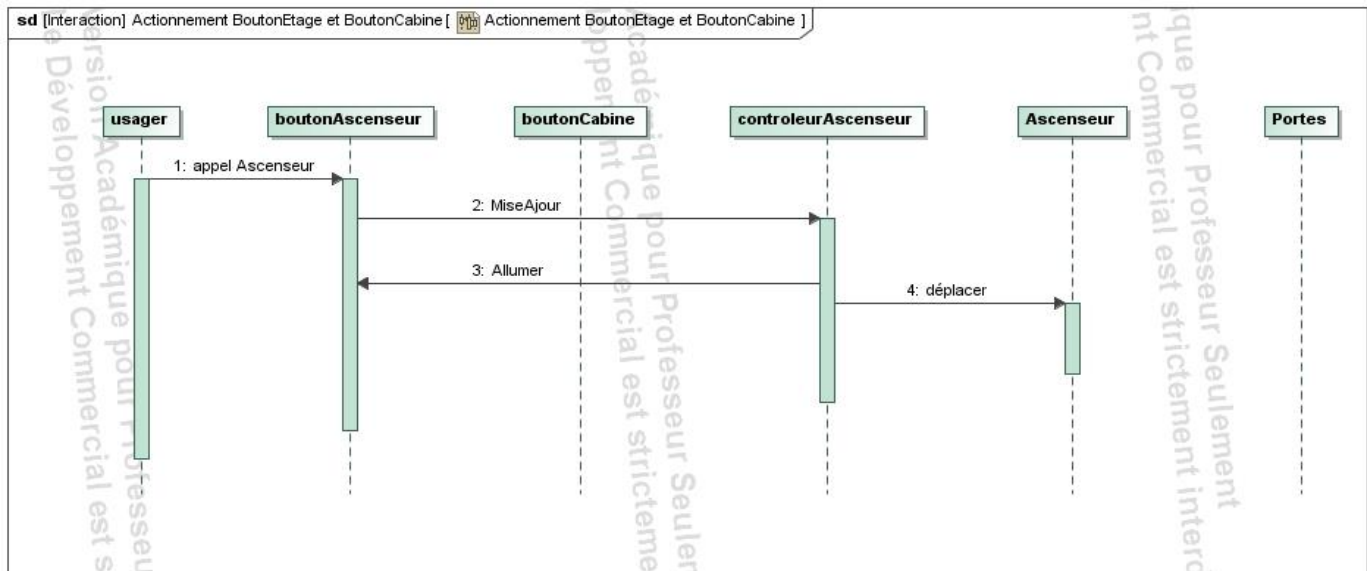
- L'utilisateur presse le bouton d'appel de l'ascenseur
- Le système détecte le bouton d'appel pressé et le voyant s'allume
- L'ascenseur bouge à l'étage demandé
- le voyant s'éteint
- Les portes de l'ascenseur s'ouvrent (temporisation)
- L'utilisateur rentre dans l'ascenseur et presse le bouton d'étage précisant l'étage demandé, le voyant s'allume
- Les portes de l'ascenseur se ferment
- L'ascenseur va à l'étage demandé
- le voyant s'éteint
- Les portes de l'ascenseur s'ouvrent (temporisation)
- L'utilisateur sort de l'ascenseur
- Les portes de l'ascenseur se ferment

Le scénario de base est représenté sous forme de diagramme de séquence simple à deux objets. Seuls sont tracés les messages entre l'utilisateur et l'ascenseur.



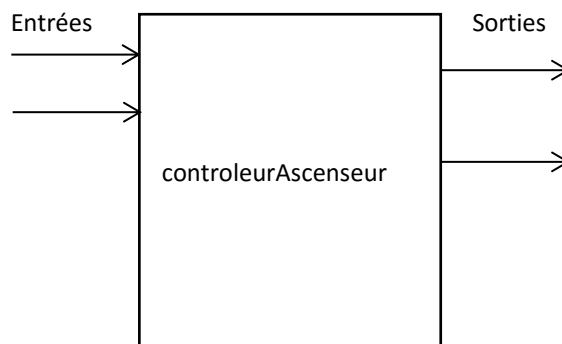
#### 3.4 DIAGRAMME DE SEQUENCE RESULTANT DU SCENARIO DE BASE

**Etape 1.** Proposer un diagramme de séquence faisant intervenir l'ensemble des objets participants à l'exécution du cas d'utilisation (temporisation de 10s pour le maintien des portes ouvertes) :



### 3.5 DETERMINATION DES ENTREES SORTIES DU **controleurAscenseur**

Le **controleurAscenseur** pour gérer le fonctionnement normal de l'ascenseur reçoit des informations d'entrée des éléments définis précédemment. Ces informations sont analysées et traitées. Le **controleurAscenseur** génère alors des ordres de sortie à exécuter par ces mêmes éléments.



**Etape 2.** Lister les entrées et les sorties du « **controleurAscenseur** ». Préciser la nature des informations reçues et envoyées.

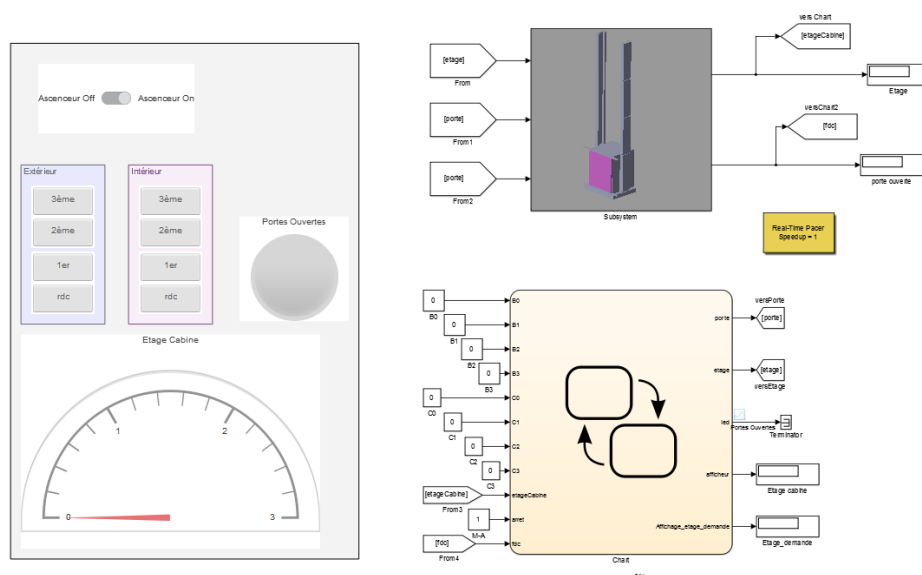
## 4 PARTIE II : ÉTUDE DE LA COMMANDE D'ASCENSEUR

### 4.1 ÉTUDE DE LA COMMANDE D'UN ASCENSEUR SIMPLE CABINE SANS MEMOIRE (UN SEUL APPEL GERE)

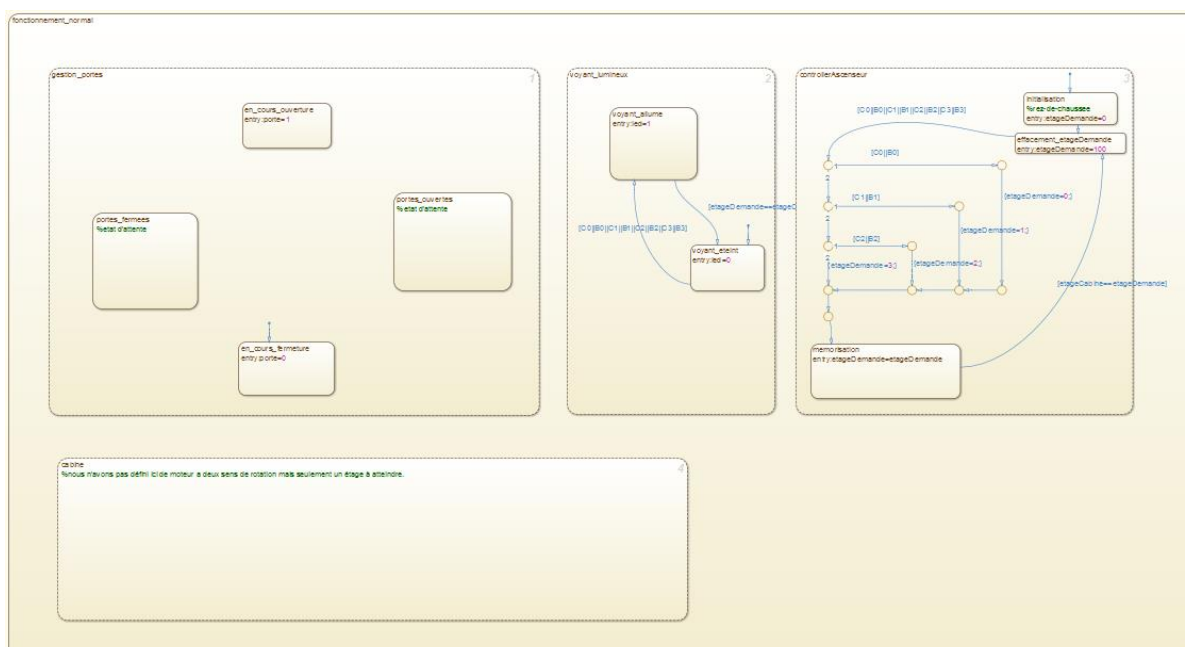
L'ascenseur étudié présente :

- Une cabine (Ascenseur) pouvant se déplacer sur 4 étages (0 à 3 inclus) ;
- Un système de double-portes (pas de porte intérieure automatique) ;
- Sur chaque palier, un bouton poussoir d'appel avec une led informant que la demande a été enregistrée : B0, B1, B2 et B3 ;
- Dans la cabine, un pupitre de commande à 4 boutons poussoirs de demande d'étage : C0, C1, C2 et C3 ;
- Une temporisation pour gérer le temps d'ouverture des portes (par exemple 10s) ;
- Un voyant led bicolore rouge et vert informant de la fermeture des portes (fermées : rouge, ouvertes : vert).
- Un capteur fin de course **fdc** qui détecte l'ouverture complète des portes ;
- Pas de capteur fin de course à la fermeture des portes (mettre une temporisation).

Les entrées et les sorties de la commande de l'ascenseur sont déjà renseignées dans le fichier « ascenseur\_simple\_boutons\_eleve ».



Les diagrammes d'états du fonctionnement normal sont donnés sauf ceux de commande de la gestion des portes (donné partiellement) et du déplacement de la cabine.



**Etape 3.** Lire et analyser les diagrammes d'état « voyant\_lumineux » et « controllerAscenseur ».

Remarque : un seul voyant est « programmé » et il simule l'ensemble des 8 voyants.

**Etape 4.** Compléter les diagrammes d'état de la « gestionportes » et « cabine ». Les états correspondants au fonctionnement des portes sont donnés, vous devez définir les transitions. Pour le fonctionnement de la cabine, vous devez définir ses états et ses transitions et faire afficher l'étage atteint sur l'afficheur.

**Etape 5.** Tester vos diagrammes et **sauvegarder** votre travail.

## 4.2 ETUDE DE LA COMMANDE D'UN ASCENSEUR SIMPLE CABINE OPTIMISE AVEC MEMOIRE

L'ascenseur étudié présente :

- Une cabine (Ascenseur) pouvant se déplacer sur 4 étages (0 à 3 inclus)
- Un système de double-portes (pas de porte intérieure automatique)
- Sur chaque palier (excepté rez-de-chaussée et dernier étage), 2 boutons poussoirs d'appel pour la montée ou la descente avec une led informant que la demande a été enregistrée : B0m, B1m et B1d, B2m et B2d et B3d
- Dans la cabine, un pupitre de commande à 4 boutons poussoirs de demande d'étage : C0, C1, C2 et C3
- Une temporisation pour gérer le temps d'ouverture des portes (par exemple 10s)
- Un voyant led bicolore rouge et vert informant de la fermeture des portes (fermées : rouge, ouvertes : vert).
- Un capteur fin de course **fdc** qui détecte l'ouverture complète des portes
- Pas de capteur fin de course à la fermeture des portes (mettre une temporisation)

Les entrées et les sorties de la commande de l'ascenseur sont déjà renseignées dans le fichier « ascenseur\_simple\_boutons\_memoire\_eleve ».

**Etape 6.** Proposer un algorithme de mémorisation et de gestion des appels.

**Etape 7.** Lire l'état controllerAscenseur et vérifier que l'on mémorise les paliers avec P0 P1 P2 P3. Compléter l'état cabine afin d'avoir priorité sur P0 puis sur P1 P2 et P3. Finaliser en complétant les états gestionportes et voyant\_lumineux.

**Etape 8.** Tester le scénario suivant : cabine à l'étage 1, appel palier étage 0, descente de l'ascenseur. Simultanément appel palier 1<sup>er</sup> étage B1m, appel palier 2<sup>ème</sup> étage B2d. L'appel cabine est C3.

## 4.3 ETUDE DE LA COMMANDE D'UN ASCENSEUR DOUBLE CABINES

L'ascenseur étudié présente :

- Deux cabines (Ascenseur) pouvant se déplacer sur 4 étages (0 à 3 inclus)
- Un système de double-portes (pas de porte intérieure automatique)
- Sur chaque palier (excepté rez-de-chaussée et dernier étage), 2 boutons poussoirs d'appel pour la montée ou la descente avec une led informant que la demande a été enregistrée : B0m, B1m et B1d, B2m et B2d et B3d
- Dans chaque cabine (droite et gauche), un pupitre de commande à 4 boutons poussoirs de demande d'étage : C0d, C1d, C2d, C3d et C0g, C1g, C2g, C3g

Le fichier matlab « ascenseur\_double.slx » présente la partie simulink de l'ascenseur à deux cabines.

**Etape 9.** Définir les éléments nécessaires à la réalisation de la commande de l'ascenseur.

Un ascenseur peut être programmé avec une priorité à la montée ou une priorité à la descente.

**Etape 10.** Proposer un algorithme présentant la commande avec priorité à la descente.