C06_TP01 : Modélisation des systèmes à événements discrets

Machines à états

Table des matières

Problématique

On se propose dans ce TP/TD de modéliser et simuler le fonctionnement d'un système, à événements discrets, simple.

La modélisation et la simulation du fonctionnement sera réalisée à l'aide du logiciel **Yakindu**.

Vous pouvez télécharger le logiciel soit la version professionnelle (for academic use) ou la version standard (qui devrait suffire). https ://info.itemis.com/state-machine/download-yakindustatechart-tools

Une vidéo précise la procédure d'installation https ://youtu.be/UmdQPZvP5x4

Vous devez avoir sur votre système une machine java https ://www.java.com/fr/download/ installée pour que le logiciel fonctionne.

Description du système (Lampe automatisée)

Cahier des charges

Le fonctionnement d'un éclairage extérieur est décrit par le diagramme ci-contre.

— À l'état initial l'éclairage est éteint.

— Un appui sur le bouton éclaire.

— Un nouvel appui sur le même bouton éteint.

Ce diagramme comporte deux états, un pseudo état (l'état initial) et deux transitions avec un événement associé. Lorsqu'un état est actif, les actions associées sont réalisées. Le passage d'un état actif à un autre n'est possible que lorsque l'événement associé à une transition paraît.

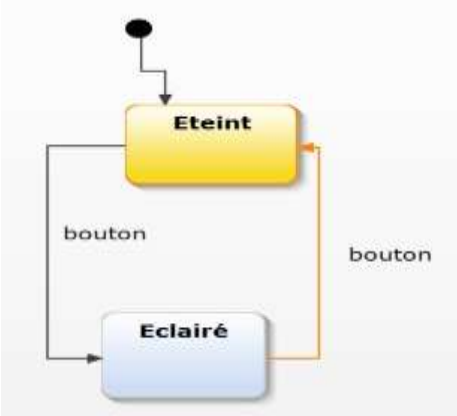


FIGURE 1 – Diagramme d'états du fonctionnement

Création des diagrammes

- Lancer Yakindu
- Créer le projet et le dossier de travail pour cela suivre la procédure décrite dans la vidéo suivante.
- Construire le diagramme

Ce diagramme simple comporte deux états (Éteint/Éclairé), un pseudo état (état initial) et deux transitions.

Pour les besoins de la simulation, l'événement « **bouton »** est un événement interne.

Yakindu permet de simuler le fonctionnement en animant l'évolution des états.

Pour pouvoir créer son premier diagramme d'états cliquez sur le lien : Premier_diagramme_detat

Cahier des charges : Fonctionnement temporisé

Le fonctionnement initial est conservé mais si l'éclairage fonctionne plus de x secondes, il s'éteint automatiquement.

Le graphe ci-contre, Figure 2, décrit ce fonctionnement.

Le mot clef « **after** » est un mot réservé du langage qui déclenche un événement interne lorsque le délai est écoulé.

Il suffit simplement de rajouter une transition entre l'état éclairé et l'état éteint en rajoutant l'évènement « after 3s ».

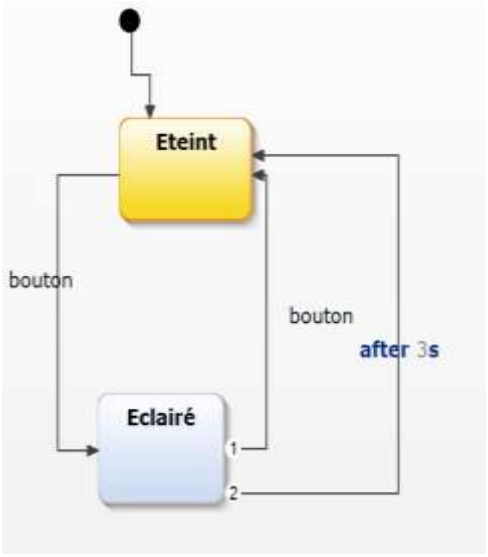


FIGURE 2 – Fonctionnement temporisé

Le cahier des charges évolue.

Cahier des charges : Intensité Variable

On ajoute la possibilité d'augmenter la luminosité à la demande de l'utilisateur.

On commence par ajouter une **interface** utilisateur qui permettra de communiquer par la suite avec l'extérieur.

Dans cette interface, on précise deux événements d'entrée **in event bouton_on** et **in event bouton_off** et une variable de type entier **var brillance:integer**

Ce fonctionnement est décrit sur la figure 3.

La vidéo https://youtu.be/GS5WaY8sUNk décrit la procédure de ce qui va suivre.

On remarque dans ce diagramme d'autres particularités du langage.

Une transition peut :

- Ne comporter qu'un **événement** : **user.bouton_off et after 10s**. Si l'état précédent est actif et que l'événement est vrai, alors l'état est désactivé et l'état suivant est activé.
- Comporter un **événement** et une **action** : **user.bouton_on/user.brillance=1**. Si l'état précèdent est actif et que l'événement est vrai, alors la transition est franchie et pendant le franchissement, l'action associée est réalisée.

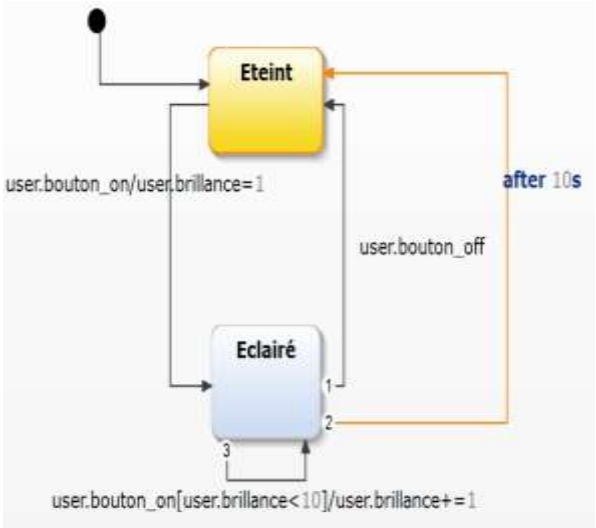
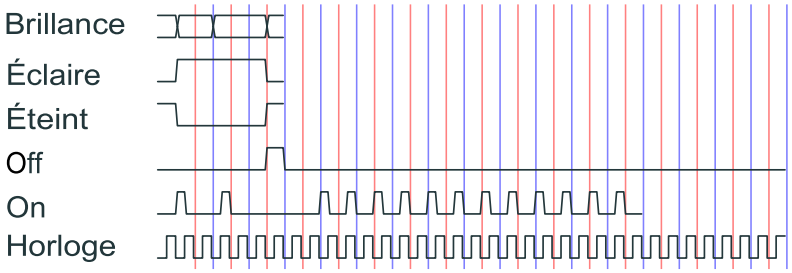


FIGURE 3 – Fonctionnement avec intensité lumineuse variable

- Comporter un **événement**, une **condition** et une **action** : **user.bouton_on[user.brillance<10]/user.brillance+=1**.
- Le passage d'un état au suivant ne peut avoir lieu que si l'événement est vrai ainsi que la condition booléenne.

À partir de la simulation compléter le chronogramme ci-dessous (la temporisation est réglée sur 4s) :



Cahier des charges : Détecteur de présence

On considère maintenant que l'éclairage est équipé d'un détecteur de présence.

- Au **repos** l'éclairage est éteint.
- Si le **capteur de présence** détecte un mouvement, la lumière s'éclaire.
- Au bout de **30 s** l'éclairage s'éteint si aucun mouvement n'est détecté, dans le cas contraire, la temporisation redémarre.
- L'utilisateur a la possibilité de forcer en manuel l'éclairage en appuyant sur **le bouton_on**.
- L'utilisateur peut ajuster l'intensité en appuyant plusieurs fois sur le même bouton.
- La sortie du mode manuel intervient soit au bout de 10 s soit en appuyant sur **bouton_off**.

Le diagramme de la figure 4 décrit ce fonctionnement.

On trouve dans ce diagramme de nouveaux éléments du langage.

Un état **composite** : l'état **Automatique**, celui-ci comporte un diagramme dans l'état. On peut ainsi décrire un système en structurant l'étude d'un niveau le plus large vers un niveau plus précis. On remarque l'état initial qui est actif à l'instant de l'entrée dans l'état.

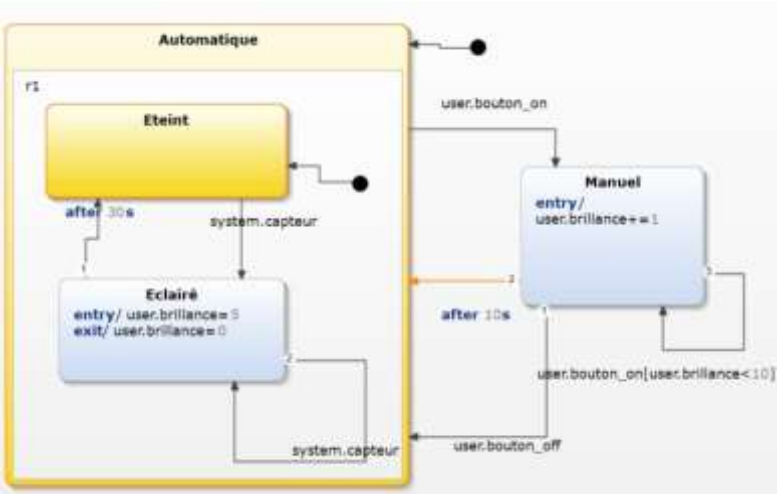


FIGURE 4 – Fonctionnement avec un détecteur de mouvement

Les deux instants

- **entry /** : les actions associées à cet instant sont réalisées inconditionnellement à l'entrée dans l'état : **entry / user.brillance=5**, à l'entrée dans l'état l'éclairage est allumé.
- **exit /** : les actions associées à cet instant sont réalisées inconditionnellement à la sortie de l'état : **exit / user.brillance=0**, à l'entrée dans l'état l'éclairage est éteint.

La vidéo https://youtu.be/AEywYBzDoJM précise cette procédure qui vient d'être décrite.

Ajout d'un détecteur de présence avec plusieurs éclairages indépendants

La zone à éclairer comporte plusieurs espaces chacun équipé d'un détecteur qui doivent s'éclairer indépendamment les uns des autres avec des durées d'éclairage différentes.

Pour chaque espace, on retrouve donc le même fonctionnement que dans l'étude précédente. L'utilisateur pouvant bien sûr éclairer simultanément les deux zones.

Le fonctionnement est décrit par le diagramme de la figure 5.

On trouve ici, dans l'état **Automatique** deux diagrammes qui décrivent chacun le fonctionnement d'une des lampes. Les deux diagrammes sont chacun dans une région. Les deux diagrammes sont dits **orthogonaux**.

Chaque diagramme est activé à l'activation de l'état encapsulant puis chacun se déroule selon l'évolution des états. La désactivation de l'état encapsulant désactive les deux diagrammes.

Ajout d'un détecteur de présence

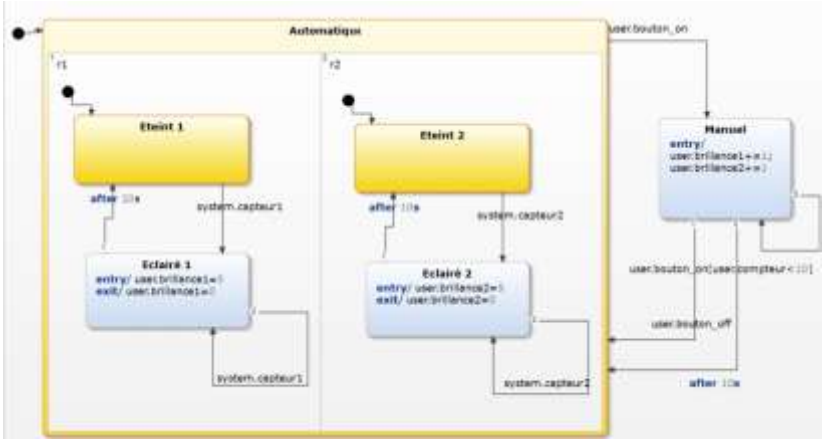


FIGURE 5 – Fonctionnement avec plusieurs éclairages indépendants