

Durée : 1 Heure

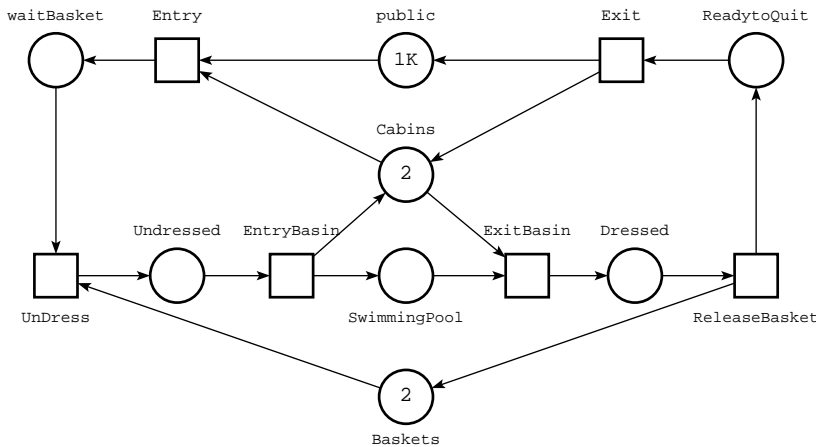
Support de cours autorisé

Les 2 exercices sont indépendants. Barème indicatif et non contractuel

1 Contexte de l'examen

On veut modéliser en Uppaal le problème de la piscine vu dans la partie Petri.

Vous travaillerez au choix en textuel ou en graphique mais n'oubliez pas de bien préciser les déclarations locales ou globales, les "interfaces" des process, ...



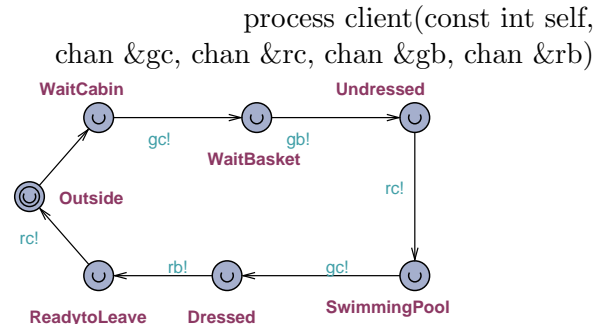
Une piscine comporte c cabines pour se changer et p paniers pour déposer ses vêtements (ici $c = p = 2$). On n'entre dans la piscine que si une cabine est libre. On attend un panier pour se changer et déposer ses vêtements. On libère la cabine et on pénètre dans le bassin. On ne quitte le bassin que si une cabine est libre. On se change et on restitue panier puis cabine en quittant la piscine.

1.1 Début de modélisation (7/20)

Questions : (2+3+2)

On veut effectuer une modélisation Uppaal du système ci-dessus en considérant deux types de process : un process "client" dont le comportement est représenté ci-contre et un process "gestionnaire" permettant de gérer les ressources (paniers ou cabines suivant instanciation). On supposera disposer de deux cabines et de deux paniers ;

On vous basant sur le comportement fourni pour le process "client", complétez la description Uppaal; Vous devrez :



1. définir les variables globales du système
2. donner le process gestionnaire (interface, variables locales, comportement),
3. Vous proposerez une instanciation du système comportant deux instances du process "client" et deux instances du process "gestionnaire" respectivement dédiées à la gestion des cabines et à la gestion des paniers.

1.2 Spécification UPPAAL (8/20)–(3+5))

1. En utilisant la logique UPPAAL, exprimez les propriétés suivantes :
 - (a) le nombre de cabines libres est toujours supérieur strictement à zéro
 - (b) un client dans le bassin quittera inévitablement la piscine
 - (c) il existe une exécution du système où il y a toujours une cabine disponible
2. Considérant la configuration de la section 1 - i.e. système instancié par deux clients, un gestionnaire de deux paniers et un gestionnaire de deux cabines
 - (a) Dites si les 3 propriétés précédentes sont satisfaites.
 - (b) En conservant la même instanciation pour les deux gestionnaires, quel est le nombre minimum d'instances de clients permettant de bloquer le système ?
Quelles sont alors, parmi les 3 propriétés précédentes, celles qui restent vraies.

1.3 Extension du modèle (5/20)

On désire ajouter un module de contrôle permettant d'enregistrer l'identité des clients ; tout usager doit décliner son identité avant d'entrer dans le bâtiment de la piscine et tenter de se procurer cabine ou panier. De même, il signale son départ de la piscine au contrôleur. Le contrôleur enregistre les clients présents dans le bâtiment.

Proposez un nouveau process “`contrôleur`” permettant de mettre en oeuvre le comportement attendu. Vous indiquerez aussi les nouvelles déclarations globales ou locales nécessaires. Compte-tenu du temps alloué, il n'est pas demandé de modifier le comportement du client.

Pour simplifier, on supposera connu le nombre maximum de clients potentiels (`NbMax`) et que chacun de ceux-ci est identifié par un entier compris entre 0 et `NbMax-1`. A noter que l'identité d'un client est déjà encodée via la constante `self` dans l'interface du process.