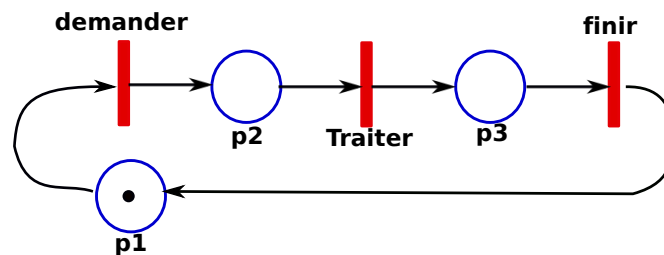


encadrés par C. Attiogbé, S. Faucou

Cahier d'exercices 1 - Modélisation avec les réseaux de Petri *Place/Transition*

Tous les exos doivent être faits (finissez à la maison ceux qui ne sont pas traités en TD)

Nombre de séances : ... Compte-rendu demandé : ...selon consignes...



Rappels

Les réseaux de Petri *Place/Transition* permettent de décrire

- le comportement de processus (sous-systèmes) et les manières d'accéder à des ressources (données) partagées par des processus ;
- le comportement global d'un système ;
- l'effet de l'accès et de la manipulation des ressources sur le système global.

Pour traiter les exercices suivants, vous allez vous baser sur les exemples déjà vus en cours pour certains (producteur/consommateur, philosophes, lecteur/rédacteur).

Méthode de modélisation

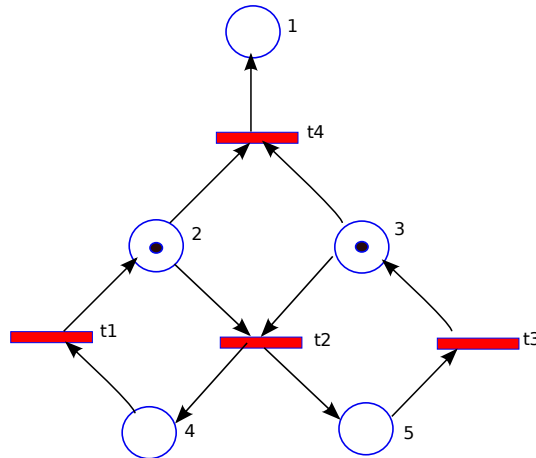
Selon le cahier de charges (ici les exercices) à traiter,

- **identifiez les ressources** et par conséquent les modes d'accès ;
chaque ressource sera systématiquement modélisée par **au moins une place** ;
- **identifiez les processus**, et par conséquent leurs comportements (suites d'actions) ; précisez pour chaque processus, quelles ressources il utilise ;
de façon générale une action est modélisée par **une transition** ; cependant pour certaines actions il faut une autorisation (représentée par **une transition**) quand les conditions pour effectuer l'action sont remplies. Ces conditions sont représentées par **des places**. Ainsi chaque action (du comportement) d'un processus est précédée d'**une transition** qui l'autorise et éventuellement suivie par une transition.
- réutilisez les techniques de base (pour l'accès à une ressource : demander, utiliser, libérer), **synchronisation, exclusion mutuelle**.
- Vous pouvez alors synthétiser avec un tableau comme le suivant :

	ressources ₁	ressources ₂	ressources ₃	...
Processus ₁	actions _i ,...	actions _j ,...	actions _r ,...	actions _u
Processus ₂	actions _j ,...	actions _i	actions _k	actions _i
...	actions _r ,...	actions _j ,...	actions _r ,...	actions _r ,...

Exercice - rappels du fonctionnement d'un réseau de Petri

Ouvrez votre support de cours et exploitez le pendant tout le TD. Soit le réseau de Petri suivant :



Q#1 Listez l'ensemble des places de ce réseau.

Q#2 Listez l'ensemble des transitions de ce réseau.

Q#3 Quelles sont les transitions *tirables* (ou *franchissables*) dans ce réseau ? Expliquez pourquoi.

Q#4 Ecrivez une bande dessinée constituée des étapes successives du fonctionnement du réseau (en tirant une transition à chaque étape).

Q#5 Donnez le marquage initial de ce réseau (en utilisant la fonction $\mu(\dots)$ comme dans le cours).

Q#6 Réduisez votre bande dessinée en utilisant les marquages successifs comme abstraction du réseau.

Exercice 1 : Producteur / Consommateur (vu en cours)

Un système producteur/consommateur met en jeu un *processus producteur*, un *processus consommateur* et une *ressource partagée* sous forme d'un tampon ; le processus producteur y dépose des produits qui sont ensuite retirés par le processus consommateur avec la contrainte de **synchronisation** exigeant au consommateur d'attendre au moins une production avant de pouvoir consommer.

Q#7 Construisez le réseau de Petri correspondant à un système producteur/consommateur avec un tampon de **taille illimitée**.

Q#8 Construisez le réseau de Petri correspondant à un système producteur/consommateur avec un tampon de **taille limitée**.

Exercice 2 : Accès à une ressource en exclusion mutuelle

Rappel du principe de l'exclusion mutuelle entre processus : à un moment donné un seul processus accède/utilise une ressource qu'il a demandée et obtenue. Après usage de la ressource, le processus la libère.

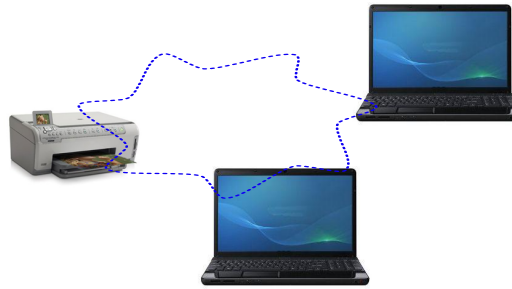


FIGURE 1 – Accès à une imprimante partagée

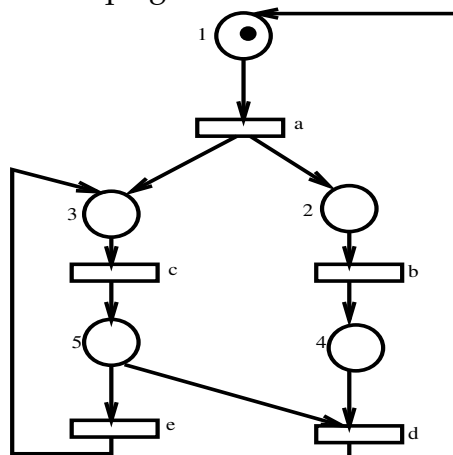
Q#9 Des usagers sur un réseau partagent entre autres, une imprimante ; ils envoient des messages pour imprimer des documents. Construisez le réseau de Petri modélisant l'accès à l'imprimante par deux processus p_1 et p_2 (représentant les usagers). On veut que les impressions de chaque processus se déroulent en exclusion mutuelle. Il n'y a aucune sorte de priorité entre les processus.

Q#10 On veut gérer l'accès à une ressource par deux processus. Ces processus peuvent être soit au repos (par rapport à la ressource), soit en demande de la ressource, soit en utilisation de la ressource.

Construisez le réseau de Petri modélisant cette gestion.

Exercice 3 : RdP, graphe, expression régulière

Q#11 Construisez le graphe de marquage associé au réseau suivant.



Q#12 Donnez une expression régulière décrivant le langage associé au réseau (ou comportement du système modélisé).

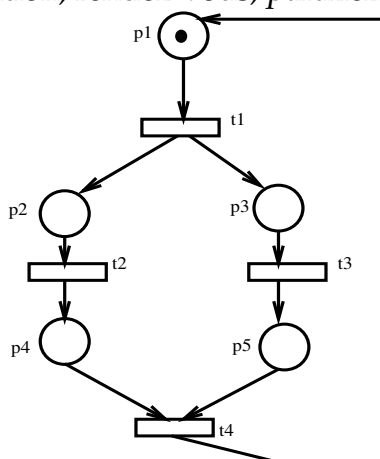
Exercice 4 : le problème des philosophes

Quatre philosophes $\phi_1, \phi_2, \phi_3, \phi_4$ partagent leur temps entre repos (pensée) et repas. Avant de manger ils doivent **se munir successivement du couvert situé à leur droite puis du couvert situé à leur gauche**. On a vu qu'on obtient un interblocage très rapidement avec cette stratégie (qui est une pouvant surgir dans le cas général).

Q#13 Modéliser à l'aide de réseau de Petri le problème des philosophes en évitant soigneusement la situation d'*interblocage*.

Exercice 5 : graphe de marquage

Q#14 Construisez le graphe de marquage associé au réseau suivant et discutez le cas échéant les aspects suivants : *synchronisation, rendez-vous, parallélisme, exclusion mutuelle*.



Q#15 Donnez l'expression régulière correspondant au comportement du réseau.

Etude de cas : surveillance automatisée d'une zone industrielle

On veut modéliser, concevoir et développer un logiciel pour le contrôle d'une zone industrielle.

Dans les systèmes de contrôle (électronique et informatique), les capteurs (*sensors*) envoient des informations de l'environnement physique au logiciel/ordinateur qui les traite, et envoie en retour des informations aux actionneurs (*actuators*) qui agissent sur l'environnement physique.

La zone industrielle est structurée en secteurs (comme sur la figure Fig. 2). Un contrôleur est associé à un seul secteur ; mais plusieurs contrôleurs peuvent contrôler un même secteur.

Le contrôle de la zone industrielle par une application répartie impliquant les contrôleurs de secteurs, consiste à récupérer des informations sur les différents secteurs de la zone, les analyser, détecter d'éventuelles anomalies, puis recommander des actions ou lever des alertes. Les informations, recueillies par des capteurs sur les différents secteurs et analysées, sont par exemple la température, l'hygrométrie, la luminosité, la pression, etc.

Une propriété attendue du logiciel est que tous les secteurs de la zone soient contrôlés (par au moins un contrôleur).

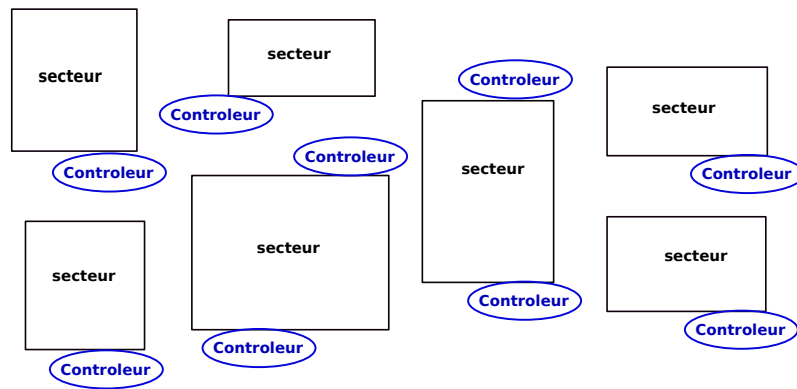


FIGURE 2 – Des secteurs dans une zone industrielle

Pour simplifier, les informations recueillies sur les secteurs sont d'une seule nature (la température) et sont des entiers naturels entre un seuil min (SeuilMin) et un seuil max (SeuilMax). Ainsi la température d'un secteur est normale, si elle est située entre SeuilMin et SeuilMax.

Pour aborder l'étude du système on le décompose et on procède par constructions successives comme suit (voir Fig. 3).

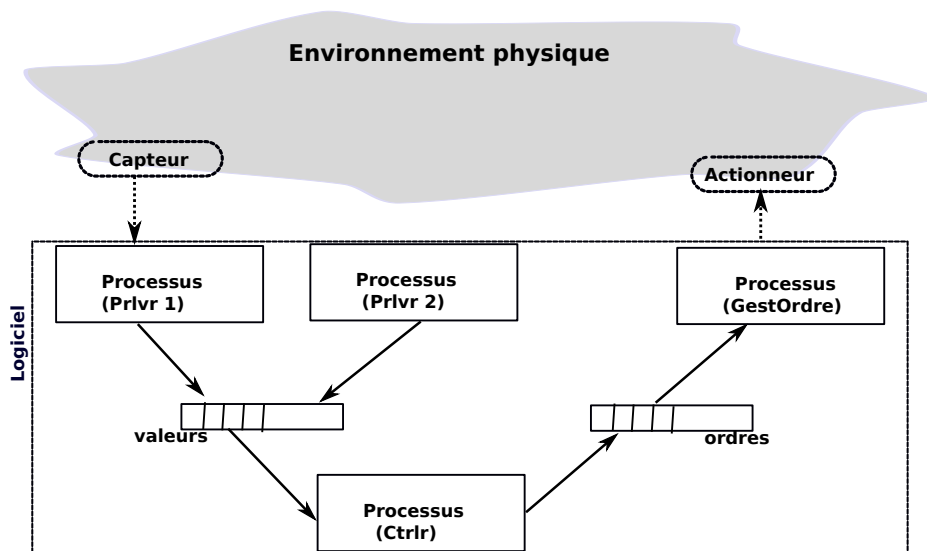


FIGURE 3 – Principe du contrôle

Deux processus (nommés Privr) doivent écrire des valeurs prélevées à partir d'un capteur, dans un registre (commun) qui est lu par un processus de contrôle (Ctrlr).

Q#16 Modélisez le comportement d'un processus (Privr) qui peut continuellement : prélever la valeur d'un capteur, puis écrire dans un registre.

Q#17 Réécrivez le modèle précédent en considérant que l'écriture dans le registre se fait en **exclusion mutuelle** avec un deuxième processus (Privr) identique. Généralisez à trois (puis n) processus identiques.

Q#18 Modélisez maintenant le comportement d'un processus de contrôle (Ctrlr) qui, continuellement : lit le contenu du registre (s'il n'est pas vide), le traite et recommence.

On veut approfondir le traitement du processus Ctrlr. Le processus Ctrlr lit périodiquement la valeur du registre. En fonction de la valeur lue, le processus Ctrlr prépare un ordre et le dépose dans une file d'attente bornée destinée à un processus (nommé GestOrdre) qui traite les ordres. Lorsque la file est pleine, le processus Ctrlr ne peut pas déposer un nouvel ordre ; il doit alors attendre. Le processus GestOrdre lui, prélève les ordres et les traite en envoyant un signal à des actionneurs. Chaque fois qu'un ordre est prélevé, une place devient disponible dans la file d'attente.

Q#19 Complétez la modélisation.

Q#20 Identifiez les processus (les pairs client/serveur) qui apparaissent dans tout le reste de notre système. Explicitez les rôles (client ou serveur) des processus les uns par rapport aux autres. Complétez la modélisation en vous appuyant sur une esquisse de l'architecture comme celle de la figure 3.

Synthèse des acquis

A l'issue de cette séquence de TD vous devez avoir acquis les compétences suivantes. Faites votre auto-évaluation.

Concepts/thèmes	Lacunes	Acquis	Maîtrise
Modélisation de l'exclusion mutuelle			
Modélisation de la synchronisation			
Modélisation du parallélisme de traitement			
Modélisation de l'interaction entre processus			
Demande d'accès à une ressource			
Graphe de comportement			
Traces d'un processus			
Analyse du comportement d'un programme/logiciel			
...			