TD 8

Objectif : Pratiquer la modélisation avec le diagramme d'états/transitions et d'activités UML.

Partie 1 : Diagramme d'états transitions

Exercice 1

Un formulaire en ligne est rempli par un utilisateur. Quand il valide son formulaire en appuyant sur le bouton ***go***, une vérification de la cohérence des données fournies est réalisée par ***validerEntrée()***. Si les informations paraissent correctes, on lui demande de confirmer, sinon on affiche les erreurs détectées et il doit remplir de nouveau le formulaire.

Modélisez ce comportement à l'aide d'un diagramme d'états transitions.

Exercice 2

Cet exercice concerne la description d'une fenêtre d'application. Nous voulons modéliser le comportement simplifié d'une fenêtre d'application, qui répond aux stimuli de trois boutons placés dans l'angle. Une fenêtre peut être dans trois états : réduite, normale, agrandie. Lorsqu'elle est réduite, elle est représentée par une icône dans la barre des tâches. À l'état normal, elle peut être déplacée et redimensionnée.
Lorsqu'elle est agrandie, elle occupe toute la surface disponible de l'écran et ne peut être déplacée ou redimensionnée. Les arcs portent une étiquette indiquant le nom de l'événement qui déclenche la transition associée. Une nouvelle instance de fenêtre sera initialisée à l'état créée, dans le sous-état ouverte et dans le sous-état normale. Nous voulons permettre à la fenêtre de retrouver son état précédent (normale ou agrandie) quand on quitte l'état réduite (utilisation du pseudo-état historique).

Modélisez ce comportement à l'aide d'un diagramme d'états transitions.

Exercice 3

Une porte munie d'une serrure offre les opérations ouvrir, fermer, verrouiller, déverrouiller et franchir. La serrure peut être fermée à simple ou à double tour.

1. Commencez par modéliser les états et transitions de la serrure. Mettez en avant les états où la serrure est verrouillée par rapport aux états où elle ne l'est pas.
2. Exprimez à travers un diagramme de protocole la façon normale de se servir d'une porte à verrou.
3. Modélisez les états et transitions d'une porte sans serrure. Ajoutez ensuite des annotations exprimant des contraintes pour lier ce diagramme à l'utilisation de la serrure.

Partie 2 : Diagramme d'activités

Exercice 4

Cet exercice concerne la description de la Fabrication d'un produit manufacturé. Les pièces nécessaires à l'assemblage sont produites séquentiellement par l'activité ***Fournir pièce***. Dès qu'une

pièce est prête, elle peut être montée, par l'activité ***Monter pièce***, en parallèle, on s'occupe de fournir la pièce suivante si toutes les pièces n'ont pas encore été fournies. Quand il ne reste plus de pièce à fournir, le flot se termine. L'activité ***Monter pièce*** peut avoir des durées variables ; chaque fois qu'elle se termine, on teste si le montage est terminé ou non. Une fois la dernière pièce montée, le produit est emballé et l'activité englobante se termine.

Modélisez ce comportement à l'aide d'un diagramme d'activités.

Exercice 5

Cet exercice concerne la description d'un distributeur de billets. Il permet d'illustrer différentes représentations des alternatives.
Le client insère sa carte, et après la vérification préliminaire, deux activités sont potentiellement déclenchées : la restitution de la carte si la vérification échoue ou l'activité taper PIN (*Personal Identification Number*) sinon. Après le contrôle du code PIN, on trouve trois alternatives. Si le code est faux, dans ce cas-là, le client doit retaper son PIN (trois fois maximum). Si le code est toujours faux (plus de trois fois), dans ce cas-là, la carte sera avalée. Si le code est bon, dans ce cas-là, le client va saisir le montant voulu, le système vérifie qu'il a le solde pour cette opération, pour lui délivrer les billets ensuite. Sinon, la carte sera restituée au client.

Modélisez ce comportement à l'aide d'un diagramme d'activités.

Exercice 6

Cet exercice concerne la description d'un traitement Passer Commande. On identifie trois intervenants dans ce traitement, à savoir le **client**, le **service comptable** et le **service livraison** (c a d trois « couloirs d'activités ou *swimlanes* » sont utilisées, qui permettent de situer les actions par rapport aux entités du système).
Le client commence par passer une commande, ce qui donne lieu à l'élaboration d'un devis par le service comptable. L'élaboration du devis est elle-même décomposée en deux sous-activités : vérifier la disponibilité des produits commandés et calculer le prix de la commande.
Nous voulons mieux faire apparaître la transmission du devis au client (on fait figurer un nœud d'objets). Cela représente un flux de données échangées entre le service comptable et le client. ***Devis*** est le nom d'une classe du système : les flots de données sont typés. Le client peut ensuite décider de modifier sa commande (retour dans l'activité initiale ***Passer commande***), de l'annuler (passage dans l'état final, signifiant la fin de ce traitement), ou de valider le devis. S'il valide, deux actions peuvent être engagées **en parallèle** : la préparation de la commande par le service livraison et le traitement de la facturation et du paiement. Le service comptable crée une facture qu'il envoie au client (l'objet facture transmis est alors dans l'état **émise**). Celui-ci effectue le paiement et renvoie la facture (qui se trouve à présent dans l'état **réglée**). Quand la commande est prête et le paiement du client confirmé (**synchronisation** par rendez-vous de ces deux activités), la commande est livrée au client et le traitement s'achève.

Modélisez ce système à l'aide d'un diagramme d'activités.