

Chapitre 5 : Diagrammes états-transitions

Les exercices qui suivent ont pour but de permettre aux apprenants la maîtrise des concepts sous-jacents du chapitre intitulé Diagrammes états-transitions, tant du point de vue de la technique (quelles notations, comment...) que de celui de l'utilisation (dans quel contexte, pourquoi...). Ils sont (en principe) originaux. Des corrigés sont disponibles.

Exercice 5.1

Nous allons tenter de décrire la vie des étudiants de la MIAGE de Nantes, exprimée à travers d'un ensemble d'états et de transitions entre ces états. La scolarité dure trois ans. Nous allons, dans un premier temps, associer un état à chaque année. Nous appellerons ces états *EnPremièreAnnée*, *EnDeuxièmeAnnée* et *EnTroisièmeAnnée*. Plusieurs événements peuvent survenir durant le parcours des étudiants. Voici ceux qui ont été recensés : *Admission* (qui conditionne le passage dans l'année supérieure ou la délivrance du diplôme final), *Redoublement* (qui maintient l'étudiant au même niveau), *Exclusion* (qui interromp la scolarité de l'étudiant), *Sélection* (pour l'entrée en première année) et *Abandon*. Ce dernier événement, qui peut survenir n'importe quand durant la formation, entraîne la disparition de l'étudiant de la MIAGE.

- a) En utilisant exclusivement les états et transitions mentionnés dans le texte, décrivez par un diagramme états-transitions non hiérarchique la vie d'un étudiant de la MIAGE.
- b) Simplifiez ce schéma en ayant éventuellement recours à l'ensemble des concepts et notations associés par UML à ce type de diagramme.
- c) Représentez le montant du jury qui, à la suite de l'examen des résultats, prononce l'admission, le redoublement ou l'exclusion. De façon à simplifier le travail à faire, on supposera que le jury opère de façon mécanique :
- moyenne inférieure strictement à 8/20 = Exclusion
 - moyenne supérieure ou égale à 8/20 et inférieure strictement à 10/20 = Redoublement
 - moyenne supérieure ou égale à 10/20 = Admission

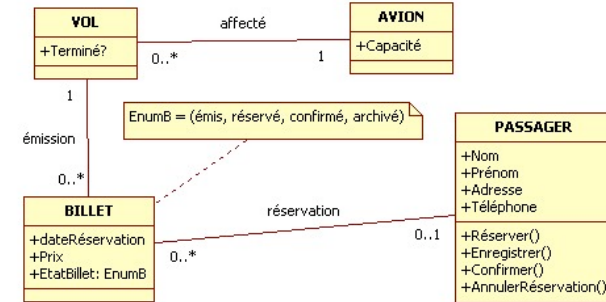
Exercice 5.2

La compagnie aérienne AirForm nous propose de modéliser en UML une version réduite de son système d'information de réservation de billets de vols. Les vols sont planifiés à l'avance et on leur affecte un avion, un aéroport de départ et d'arrivée, une date de départ et une date d'arrivée. Chaque avion dispose d'une capacité en nombre maximal de passagers. Les billets sont émis pour chaque vole lors de la planification. Il n'y a pas de surbooking (de sur-réservation).

Les usagers achètent des billets. Cet achat entraîne une réservation (via le billet) pour le vol en question. On conserve les nom, prénom, adresse et téléphone des usagers ayant fait une réservation, ainsi que la date de réservation et le prix du billet. Lors de leur enregistrement (au moment du départ), les passager confirment leurs billets pour le vol enregistré. On mémorise cette confirmation.

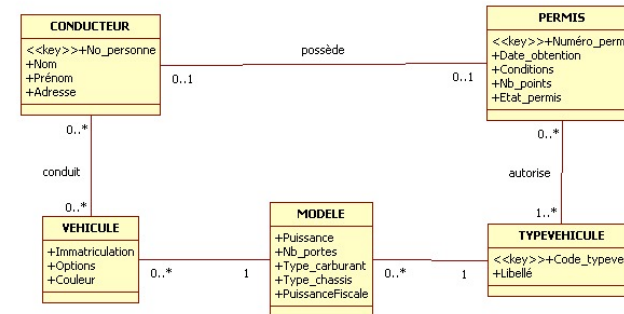
Lorsque le vol est terminé, on archive les réservations qui le concernaient. Si le vol est annulé, les réservations sont annulées.

Un premier diagramme de classes a été établi (voir page suivante). Modélisez l'évolution de l'état d'un billet par un diagramme états-transitions hiérarchique. Ce diagramme sera cohérent avec le diagramme de classes, celui-ci pouvant (devant) être modifié pour ce faire.



Exercice 5.3

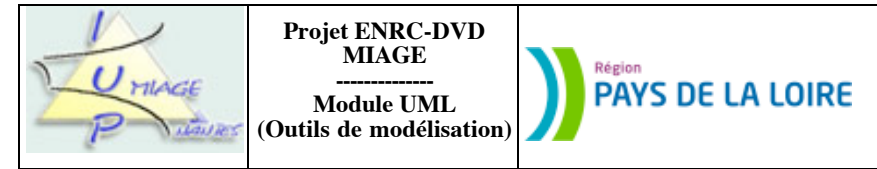
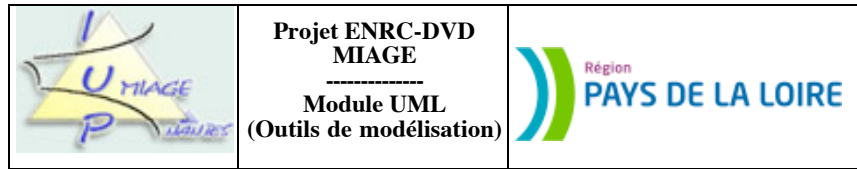
Soit le diagramme de classes suivant, modélisant une partie d'un système de gestion de permis de conduire. L'état du permis peut être : non attribué, attribué (au conducteur lié à l'association *possède*), suspendu (si le nombre de points du permis est à 0) ou supprimé (ie. archivé). Représentez cela par un diagramme états-transitions. Les deux diagrammes, états-transitions et classes, devront être mis en cohérence.



Exercice 5.4

Les collectionneurs de timbres qui souhaitent compléter leur collection peuvent recourir à des ventes sur offres, organisées un peu partout dans le monde par des marchands de timbres. En voici le principe :

Les vendeurs d'une lettre ou d'un timbre contactent le marchand et fixent avec lui le prix de départ auquel il ou elle va être mis en vente.



Exercice 5.5

Le marchand attend d'avoir suffisamment de lots (timbre et/ou lettre appartenant à un vendeur) pour constituer un catalogue de vente qui décrira minutieusement les lots et annoncera les prix de départ. Ces catalogues seront envoyés à tous les acheteurs potentiels connus du négociant.

Les collectionneurs intéressés vont faire des offres d'achat. Le prix d'achat proposé doit obligatoirement être supérieur ou égal au prix de départ. La vente se déroule selon le principe des ventes aux enchères, le plus offrant (celui qui propose le prix d'achat le plus élevé) remportant le lot.

La vente a lieu uniquement par correspondance. Elle est clôturée à une date fixée à l'avance (elle figure sur le catalogue). Lorsque la vente est close, le marchand regarde, pour chaque lot, quelle est l'offre la plus haute. Le collectionneur qui la émise est déclaré vainqueur. C'est lui qui va pouvoir acheter le lot. Il reçoit une facture et a dix jours pour la régler. Il peut se désister. S'il renonce, c'est la personne qui a émis l'offre juste en dessous qui est à son tour déclaré vainqueur.

S'il n'y a pas d'offre (le timbre n'intéresse personne) ou si la vente ne peut pas se faire (il n'y avait qu'une offre et l'acheteur se désiste), le lot est renvoyé au vendeur.

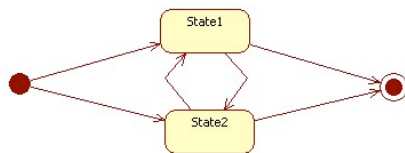
a) Un lot passe par les états suivants :

- *Avendre* : vendeur et marchand sont d'accord sur le prix de départ ; le lot n'est pas encore dans le catalogue.
- *EnVente* : le lot est décrit dans le catalogue.
- *Demandé* : un collectionneur (au moins) a fait une proposition de prix.
- *Vendu* : la vente est close ; un collectionneur est déclaré vainqueur.

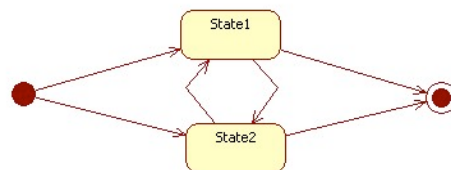
Décrivez l'évolution de la vie d'un lot à l'aide d'un automate non hiérarchique. Seuls les états mentionnés ci-dessus devront être utilisés. Les actions à prendre en compte et à placer sur les transitions sont les suivantes : *EditerCatalogue, EnregistrerOffre, EnvoyerFacture, RenvoyerLotVendeur, ExpédierLotAcheteur*.

Les événements pris en compte sont les suivants : *Lancement* (de la vente), *Clôture* (de la vente), *Offre* (d'achat), *Désistement* (d'un acheteur), *FinVente, RéceptionPaiement*.

b) Une offre passe, elle aussi, par plusieurs états. Recensez les états, les actions et les événements qui entreront dans l'automate décrivant la vie de cette offre. Les événements appartiendront obligatoirement à la liste fournie à la question précédente. Vous utiliserez le squelette d'automate suivant :



c) Décrivez la vie d'un collectionneur effectuant une offre, sur le même moule que les deux automates précédents. Vous utiliserez le squelette d'automate suivant :



La société LOCATION SA est une PME ayant pour objet la location de véhicules automobiles. Lorsqu'un client arrive à l'accueil, l'hôte l'interroge puis lui propose un choix de véhicules disponibles ; Après avoir choisi un véhicule, le client fournit des éléments d'identification et un contrat de location est établi. S'il y a lieu, le client verse un chèque de caution, dont le montant est inscrit sur le contrat. Les clés du véhicule et un exemplaire du contrat sont remis au client. Celui-ci prend possession du véhicule auprès du responsable du parc automobile. À cette occasion, un état du véhicule est rédigé par le responsable et signé par le client. Le kilométrage du véhicule est relevé et inscrit sur le document. Le fichier Véhicules est alors mis à jour.

Au retour du client, un contrôle du véhicule est effectué (kilométrage effectué, niveau d'essence, état). Le deuxième exemplaire du contrat (celui de la société) est recherché, puis l'hôte procède au décompte définitif et encaisse le montant de la location. La caution est rendue au client si le contrôle a donné satisfaction. Le client doit en principe effectuer le plein du réservoir avant de rendre le véhicule. Si ce n'est pas fait, le plein est fait par la société et facturé au client.

Au cours de ces différentes phases, toutes les informations clients sont regroupées dans un dossier. Une première analyse a permis de dégager les événements suivants :

- phase 1 (emprunt du véhicule) : douze événements (au moins) peuvent être mis en évidence. Il s'agit de : arrivée du client, proposition de choix, choix du client, établissement du contrat provisoire, remise du chèque, inscription du montant sur le contrat, remise des clés, remise de l'exemplaire du contrat, prise de possession du véhicule, rédaction de l'état du véhicule et signature de cet état, inscription du kilométrage, mise à jour du fichier.
- phase 2 (retour du véhicule) : au moins huit événements apparaissent à la lecture du texte. Il s'agit de : retour du client, contrôle du véhicule, décompte définitif, paiement, restitution de la caution, contrôle satisfaisant, réservoir vide, plein fait.

De la même façon, les opérations suivantes ont été recensées :

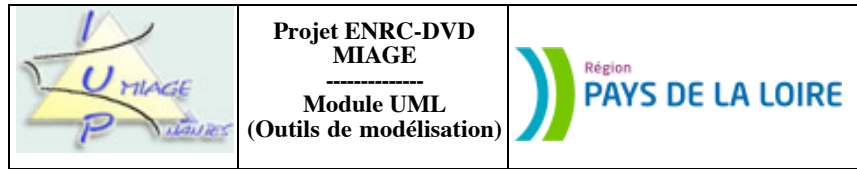
- phase 1 (emprunt du véhicule) : le travail exécuté au départ est constitué d'au moins dix opérations. Ce sont les suivantes : interroger le client, proposer un choix de véhicule, établir le contrat, compléter le contrat, remettre les clés, donner un exemplaire du contrat, prendre possession du véhicule, contrôler l'état du véhicule, noter le kilométrage, mettre à jour le fichier.
- phase 2 (retour du véhicule) : cinq opérations seulement sont nécessaires. Il s'agit des suivantes : rechercher le contrat, contrôler le véhicule, rendre la caution, encaisser le paiement, calculer ce qui est dû.

ATTENTION : cette première analyse est, peut-être, incomplète. Cela veut dire qu'il peut, peut-être, y avoir d'autres événements et/ou d'autres opérations. Cela veut également dire qu'il est possible qu'une opération recensée plus haut doit être coupée en deux ou, inversement, que deux opérations recensées plus haut doivent être regroupées en une seule.

a) Décrivez sous forme d'un automate "à plat" le comportement d'un client de cette société de location. Les cas anormaux (accident durant l'emprunt du véhicule, mauvais payeur...) ne sont pas à prendre en considération. La notation UML (celle des diagrammes états-transitions) sera adoptée.

b) Les clients de LOCATION SA peuvent bénéficier, dès la première location, d'une carte de fidélité. Celle-ci leur donne droit à 5 % de réduction sur les prochaines factures. Cette carte comporte six cases. À chaque location, une case est cochée. Lorsque la carte est pleine (ie. lorsque les six cases sont cochées), donc au bout de six locations, une septième leur est offerte. Lorsque cette location gratuite est consommée, la carte de fidélité est détruite. Le client continue à bénéficier de la réduction sur les prochaines factures. Le client recevra une nouvelle carte de fidélité lors de la huitième location. Elle sera cochée à chaque nouvelle location et donnera, lorsqu'elle sera remplie... Décrivez cette situation avec un automate "à plat". La notation UML (celle des diagrammes états-transitions) sera adoptée.

c) Peut-on combiner les deux réponses précédentes ?



Exercice 5.6

Mme V. est mandataire occasionnelle dans la société Peter GOLD, spécialisée dans la vente directe de bijoux fantaisie. Cela signifie que Mme V. se déplace à domicile, chez une personne appelée *hôtesse*, et qu'elle y effectue une présentation des bijoux à des personnes appelées *prospects*, réunies chez l'hôtesse. Ces réunions sont sans obligation d'achat. Une prospect peut ou pas de tout intéressée restera prospect. Elle peut aussi devenir *cliente*, en passant commande d'un ou de plusieurs bijoux. Intéressons-nous, dans un premier temps, aux différents types de personnes que Mme V. est amenée à côtoyer. Nous distinguons des prospects, des clientes, des hôtesse et des hôtesse potentielles (encore appelées *candidates* par Mme V.). Donnons-en une définition :

- Prospect : personne assistant à une réunion, sans l'avoir organisée, sans acheter de bijoux, sans se porter volontaire pour l'organisation d'une prochaine réunion.
- Cliente : personne assistant à une réunion et achetant un ou plusieurs bijoux.
- Hôtesse : personne organisant une réunion.
- Candidate : personne se portant volontaire pour l'organisation d'une prochaine réunion.

Cette classification peut évoluer dans le temps, une même personne pouvant passer du statut de prospect à celui de cliente, puis à celui d'hôtesse, puis, encore plus tard, revenir à un statut de cliente... Ces statuts sont liés les uns aux autres par des relations. Ainsi, une hôtesse peut fort bien acheter des bijoux et donc se comporter aussi comme une cliente. Elle peut uniquement être hôtesse (sans jamais avoir été prospect et/ou cliente) et le rester. Une hôtesse est d'abord (ceci est une règle impérative) candidate. Elle peut, un jour, devenir cliente dans une autre réunion –organisée par une autre hôtesse– ou même prospect dans cette réunion. Mme V. souhaite conserver, pour chaque personne, son nom, son prénom et son adresse complète. Compte-tenu des risques d'homonymie, elle a accepté de numéroter chaque personne mémorisée dans la base de données. Elle peut, en outre, mémoriser, pour les hôtesse seulement (candidates incluses), leur numéro de téléphone. Une personne qui vient à une réunion et qui reste prospect ne sera pas dans la base de données. Une personne déjà dans la base (elle a assisté à une réunion antérieure, au cours de laquelle elle a acheté, ou bien elle a organisé une réunion antérieure) qui vient à une nouvelle réunion et qui se comporte là comme un prospect restera dans la base.

Représentez l'évolution d'une personne sous forme d'un diagramme états-transitions.

Exercice 5.7

Un laboratoire de recherche comprend deux catégories de personnels, des administratifs et des chercheurs. Laissons tomber les administratifs et intéressons-nous aux chercheurs. Ces derniers sont, soit des chercheurs « purs », soit des enseignants-chercheurs, soit des doctorants, soit des post-doctorants. Il n'est pas possible de cumuler (il n'y a pas, par exemple, de chercheur « pur » qui soit aussi doctorant).

Les chercheurs « purs » sont rattachés à un organisme national de recherche (tel le CNRS, l'INRIA...). Les enseignants-chercheurs sont rattachés à une université ou à une école d'ingénieurs.

Les doctorants sont des étudiants inscrits dans une formation de niveau Doctorat (en première, deuxième, troisième... année). Ils sont encadrés par un ou plusieurs enseignants-chercheurs membres du laboratoire. Cet encadrement s'exprime en pourcentage, celui-ci traduisant concrètement la part prise par chaque enseignant-chercheur dans l'encadrement du doctorant. S'il n'y a qu'un seul encadrant, le pourcentage sera de 100 %.

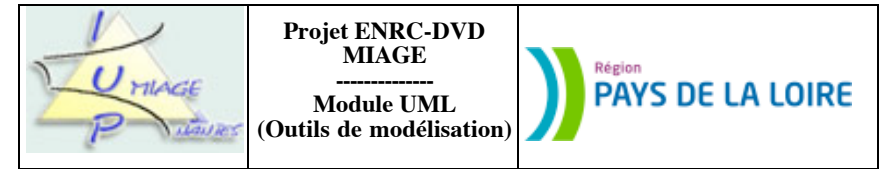
Un doctorant qui passe son doctorat devient post-doctorant. Un doctorant peut également mettre fin à ses études et abandonner la formation. Il « disparaît » alors du laboratoire.

Toutes ces personnes peuvent tomber malades (ils sont alors « en congé de maladie »). Ils peuvent aussi prendre des vacances (ils sont alors « en vacances »). Ils peuvent être envoyés en mission pour le laboratoire (leur statut est alors « en mission »). Ceux- qui ne sont ni en congé de maladie, ni en vacances, ni en mission sont « présents ».

a) Modélisez sous forme d'un diagramme états-transitions les différents statuts possibles et les transitions entre ces statuts. Vous utiliserez la notation UML de rigueur pour ce type de diagrammes et ferez particulièrement attention à bien distinguer événement, état, action.

Vous utiliserez exclusivement les vocabulaires suivants :

- pour les états : en mission, en congé de maladie, en vacances, présent ;



- pour les événements : maladie, guérison, départ en vacances, départ en mission, retour de mission, retour de vacances ;
 - pour les actions : envoyer feuille de maladie, faire compte-rendu de mission, sauvegarder données.
- Il est possible d'utiliser des gardes pour certaines transitions.

Les hypothèses complémentaires suivantes seront appliquées :

- une maladie peut se déclarer n'importe quand.
- la guérison est supposée totale. Une personne guérie peut reprendre l'activité qu'elle avait avant de tomber malade.
- il n'est pas possible de partir en vacances juste après une mission, sans revenir au laboratoire.
- il n'est pas possible de partir en vacances juste après un congé de maladie, sans revenir au laboratoire.

b) Modélisez le « passage » d'un doctorant au laboratoire sous la forme d'un diagramme états-transitions. Vous utiliserez la notation UML de rigueur pour ce type de diagrammes et ferez particulièrement attention à bien distinguer événement, état, action.

Vous utiliserez exclusivement les vocabulaires suivants :

- pour les états : primo-entrant, en cours, docteur ;
 - pour les événements : inscription, fin année, abandon, soutenance ;
 - pour les actions : délivrer carte étudiant, évaluer travail, payer un pot, rédiger démission.
- Il est possible d'utiliser des gardes pour certaines transitions.

Les hypothèses complémentaires suivantes seront appliquées :

- un post-doctorant n'est pas un doctorant.
- un doctorant ne tombe pas malade (du moins, ceci ne doit pas être exprimé dans ce diagramme).
- une soutenance de doctorat débouche toujours sur la délivrance du doctorat.
- à la fin de chaque année de doctorat, une évaluation du travail est effectuée collégialement par le ou les encadreurs. Si l'évaluation est positive, le doctorant est autorisé à s'inscrire dans l'année « supérieure » (ainsi, par exemple, un doctorant de deuxième année est autorisé à s'inscrire en troisième année). Si elle est négative, le doctorant doit quitter le laboratoire.
- un doctorant peut abandonner sa formation à n'importe quel moment. Il doit simplement remettre à son ou ses encadreurs une lettre de démission.

Exercice 5.8

Soient les deux scénarios suivants, écrits par un informaticien amateur d'objets :

S1) Emprunt effectué par un adhérent : le scénario *Emprunt effectué par un adhérent* s'exécute suivant la séquence ci-après :

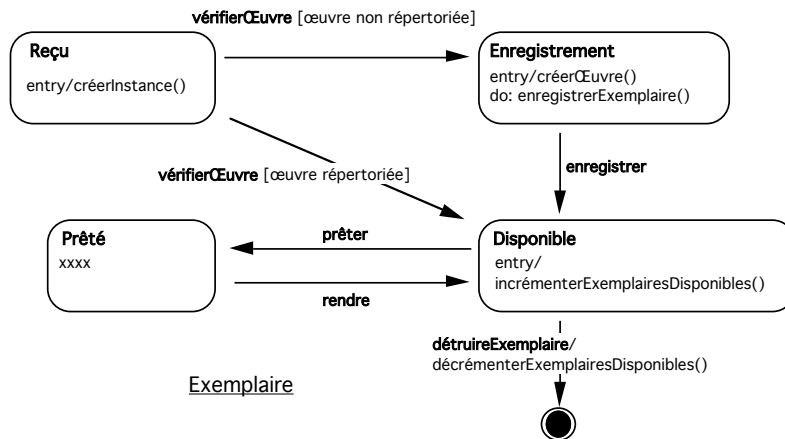
1. Un objet métier *unAdhérent* s'adresse à *unBibliothécaire*, objet externe au système, pour emprunter une œuvre : la méthode *demandeEmprunt* est impliquée dans cette action, elle appartient à la classe *Adhérent* et prend en paramètre un objet métier de type *Œuvre*.
2. *unBibliothécaire* vérifie que le nombre d'exemplaires empruntés par *unAdhérent* est inférieur à deux : la méthode d'instance *nombreExemplairesEmpruntés* de la classe *Adhérent* est déclenchée, elle renvoie le nombre d'exemplaires empruntés.
3. *unBibliothécaire* vérifie qu'il existe des exemplaires disponibles pour cette œuvre : pour cela, la méthode d'instance *exemplairesDisponibles* est envoyée à *uneŒuvre*, elle renvoie un exemplaire pouvant être prêté.
4. Le prêt pour l'exemplaire en question (*unPrêt*) peut alors être initialisé : la méthode d'instance *initialiserPrêt* est appelée, elle prend en paramètre l'objet *unExemplaire* associé.
5. *unPrêt* initialise *unTimer*, qui réveillera l'application quinze jours plus tard (voir point 1 ci-dessous).
6. *unPrêt* met à jour les données de l'objet *uneŒuvre*, en lui envoyant le message *décramerExemplairesDisponibles*.
7. *unPrêt* met à jour les données de l'objet *unAdhérent* en lui envoyant le message *incrémenterNbEmprunts*.

8. Enfin, *unBibliothécaire* attribue *unPrêt* à *unAdhérent* : la méthode *attribuer* de la classe *Adhérent* est déclenchée, elle prend en paramètre *unExemplaire*.

S2) Cas de litige avec un adhérent : le scénario *Cas de litige avec un adhérent* s'exécute suivant la séquence ci-après :

1. *unTimer* se déclenche quinze jours après son initialisation (voir point 5 ci-dessus), il notifie le litige à *unBibliothécaire*.
2. En réponse, *unBibliothécaire* met à jour *unPrêt* en lui déclenchant la méthode d'instance *miseEnLitige* de la classe *Prêt*.
3. *unPrêt* déclenche alors la méthode *envoyerAvertissement* de la classe *Prêt*.
4. *unBibliothécaire* envoie à *unAdhérent* le message *imprimerCourierLitige*.
5. *unBibliothécaire* interdit alors l'emprunt de nouveaux exemplaires à *unAdhérent*, en déclenchant la méthode d'instance *interdireEmprunt* de la classe *Adhérent*.

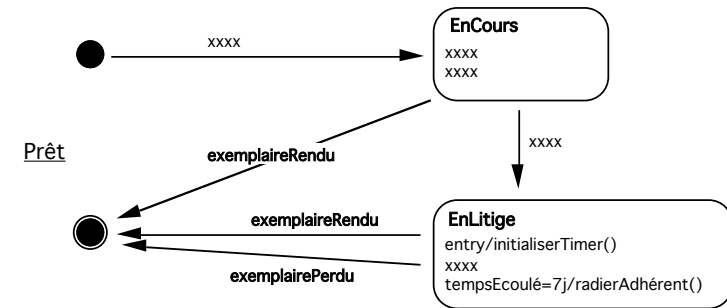
a) Complétez le diagramme d'états de la classe *Exemplaire* suivants, à partir des éléments contenus dans les scénarios (les parties à compléter sont matérialisées par des *xxxx*).



Remarque StarUML Pour réaliser ce schéma, nous avons renoncé à utiliser le logiciel StarUML. Nous avons rencontré, en effet, deux problèmes :

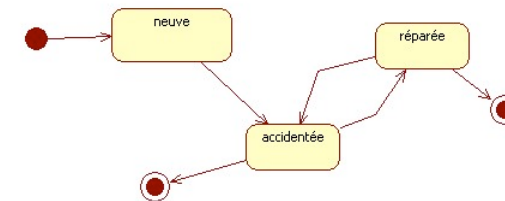
- le premier tient au fait qu'il n'est pas possible avec StarUML d'associer une opération à un événement. Les seules situations envisagées correspondent aux étiquettes *entry*, *do* et *exit* ; *on event* n'est pas prévu.
- le second est dû au type d'exercice qui est proposé. Il s'agit de remplacer des opérations anonymes, matérialisées par des *xxxx*, par des méthodes évoquées dans le texte. La pratique normale, avec StarUML, est de sélectionner une des opérations de la classe et de l'associer à la transition. Nous aurions dû « remplir » la classe d'opérations *xxxx1*, *xxxx2* etc. que nous n'avons pas souhaité faire.

b) Complétez le diagramme d'états de la classe *Prêt*, à partir des éléments contenus dans les scénarios (les parties à compléter sont matérialisées par des *xxxx*).



Exercice 5.9

Nous allons essayer de modéliser la vie d'une voiture. Une première tentative donne l'automate suivant :



a) Donnez le nom des transitions de cet automate. Vous utiliserez **exclusivement** un des événements suivants : *achat*, *accident*, *réparation*, *vente*.

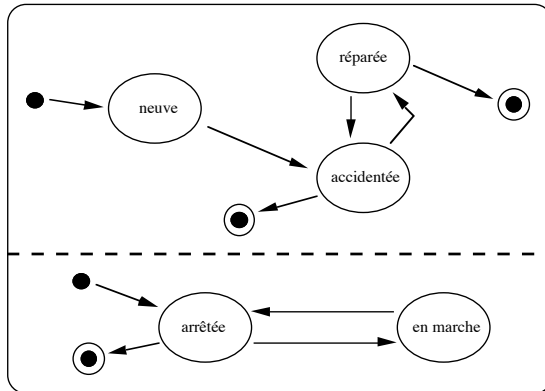
b) Ajoutez au schéma les actions suivantes : *payer la voiture*, *payer la réparation*, *appeler le garagiste*, *encaisser l'argent de la vente*.

c) Complétez l'automate précédent (celui obtenu en répondant à la question b) pour qu'il corresponde à la description suivante : une voiture neuve est sous garantie jusqu'à ce qu'elle ait fait 1500km. Elle est ensuite hors garantie. Elle peut être en bon état, accidentée ou réparée. Une voiture réparée sera dite en bon état si le contrôle effectué en fin de réparation est satisfaisant. Elle restera réparée sinon. Toute voiture (neuve, réparée...) peut avoir un accident. Elle peut aussi être vendue à tout moment.

d) Modifiez l'automate que vous avez fourni en répondant à la question précédente (la c) pour le simplifier en adoptant la notion de super-état et en utilisant intelligemment le fait qu'une voiture peut à tout moment être vendue ou avoir un accident.

e) Avec ce dernier automate, placez une transition correspondant à l'événement *utilisation* durant lequel l'action *comptabiliser les kilomètres* est exécutée.

f) L'automate suivant ne représente pas bien la réalité. Pourquoi ?



Remarque StarUML Nous n'avons pas utilisé le logiciel StarUML pour « faire » cette figure du fait du parallélisme, mais un logiciel de dessin appelé MacDraw. Il est tout à fait possible de définir avec StarUML des états concurrents (il y a une propriété prévue pour cela), mais l'effet de cette définition n'apparaît pas sur le schéma. On notera également la différence de notation entre les schémas produits. Avec StarUML, les gardes sont placées en fin d'étiquettes de transitions ; avec MacDraw, les gardes sont placées au début.

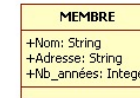
Exercice 5.10

Une association est composée de membres et d'un conseil d'administration de trente membres, dits actifs. Ce nombre peut évoluer en modifiant les statuts. L'administration de l'association est confiée à un bureau composé de membres actifs. Il est formé d'un président, de deux vice-présidents, d'un président d'honneur, d'un secrétaire et de son adjoint, d'un trésorier et de son adjoint. Toutes les fonctions doivent être attribuées pour que la préfecture accepte les statuts de l'association. Aucun membre ne peut cumuler plusieurs fonctions dans le bureau.

Le bureau est élu par le conseil d'administration, lui-même élu par l'assemblée générale, selon les règles fixées par les statuts de l'association. Actuellement, les statuts prévoient un renouvellement annuel du tiers des membres du conseil. Tous les ans, un tiers du conseil d'administration est donc déclaré sortant. Autrement dit, un membre élu l'année N est déclaré sortant l'année N + 3. L'élection d'un nouveau membre à ce conseil est soumis au vote à la majorité absolue lors de l'assemblée générale annuelle présidée par le président de l'association.

Le vote est licite si au moins les deux tiers des membres sont présents. Un membre actif ne se trouve que dans un tiers (le nombre de membres actifs est un multiple de trois). Il y a quatre possibilités de vote pour un scrutin donné : oui, non, blanc, abstention.

Étant donnée la définition du type MEMBRE, qui représente un membre de l'association, sous forme de la classe UML ci-après, définissez son évolution dans l'association par un automate. La notation UML sera utilisée.



NB : en cas de doute sur une partie du texte ci-dessus, il pourra être levé par une hypothèse.

-----fin du texte-----