

Modélisation fonctionnelle

exercices corrigés et conseils méthodologiques

Mots-clés

■ Diagramme de cas d'utilisation ■ Relations entre cas d'utilisation ■ Navigabilité des associations entre acteurs et cas d'utilisation ■ Description textuelle d'un cas d'utilisation ■ Diagramme de séquence système ■ Fragment d'interaction, opérateur ■ Diagramme d'états des opérations système.

Une nouvelle étude de cas va nous permettre dans ce chapitre de compléter le passage en revue des principales difficultés relatives à la mise en œuvre de la technique des cas d'utilisation.

Nous allons élaborer un diagramme de cas d'utilisation complexe incluant les différents types de relations entre cas d'utilisation ainsi qu'une notation avancée : la flèche de navigabilité sur les associations entre acteurs et cas d'utilisation. Nous introduirons ensuite la différence entre cas d'utilisation essentiel et cas d'utilisation réel, initialement proposée par [Larman 97], et observerons comment elle influence la description textuelle. Nous utiliserons les nouveautés les plus intéressantes du diagramme de séquence, comme les fragments d'interaction, avec les opérateurs les plus utiles. Enfin, nous donnerons un exemple de diagramme d'états montrant la séquence forcée des opérations système lors d'un cas d'utilisation particulier.

ÉTUDE D'UN TERMINAL POINT DE VENTE (TPV)



Cet exercice concerne un système simplifié de caisse enregistreuse de supermarché. Il est largement inspiré de l'étude de cas initialement proposée par [Larman 97] et qui forme la base de la formation OOAD¹ de Valtech Training.

Le déroulement normal d'utilisation de la caisse est le suivant :

- Un client arrive à la caisse avec des articles à payer.
- Le caissier enregistre le numéro d'identification (CPU) de chaque article, ainsi que la quantité si elle est supérieure à un.
- La caisse affiche le prix de chaque article et son libellé.
- Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente.
- La caisse affiche le total des achats.
- Le client choisit son mode de paiement :
 - numéraire : le caissier encaisse l'argent reçu, la caisse indique la monnaie à rendre au client ;
 - chèque : le caissier vérifie la solvabilité du client en transmettant une requête à un centre d'autorisation *via* la caisse ;
 - carte de crédit : un terminal bancaire fait partie de la caisse. Il transmet une demande d'autorisation à un centre d'autorisation en fonction du type de la carte.
- La caisse enregistre la vente et imprime un ticket.
- Le caissier donne le ticket de caisse au client.

Après la saisie des articles, le client peut présenter au caissier des coupons de réduction pour certains articles. Lorsque le paiement est terminé, la caisse transmet les informations sur le nombre d'articles vendus au système de gestion de stocks.

Tous les matins, le responsable du magasin initialise les caisses pour la journée.

1. OOAD = Object Oriented Analysis & Design.

Étape 1 – Réalisation du diagramme de cas d'utilisation



EXERCICE 2-1.

Diagramme de cas d'utilisation

Élaborez un diagramme de cas d'utilisation détaillé de la caisse enregistreuse.

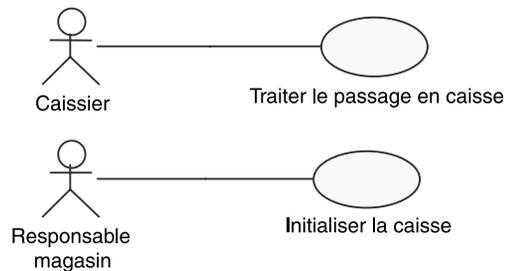
N'hésitez pas à utiliser les relations entre cas d'utilisation pour rendre votre diagramme plus précis.

solution

Dans un premier temps, une solution simpliste consiste à identifier un « gros » cas d'utilisation qui contient la totalité du déroulement normal d'utilisation de la caisse et un autre cas d'utilisation qui traite de l'initialisation de la caisse par le responsable du magasin.

Figure 2-1.

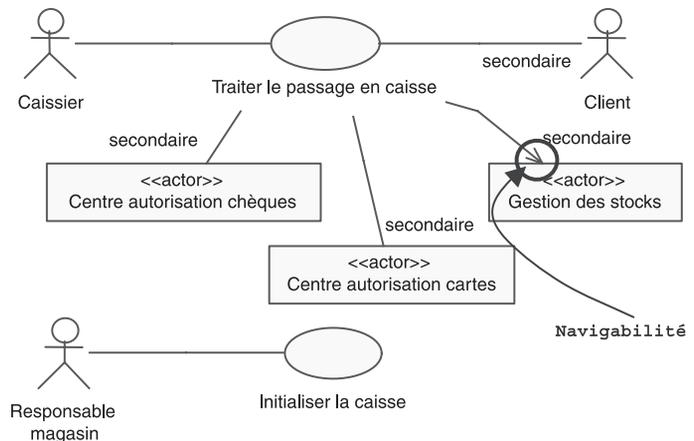
Première ébauche du diagramme de cas d'utilisation



Si l'on ajoute les acteurs secondaires sur le schéma précédent, on s'aperçoit que le cas d'utilisation *Traiter le passage en caisse* communique avec un grand nombre d'acteurs différents.

Figure 2-2.

Deuxième ébauche du diagramme de cas d'utilisation



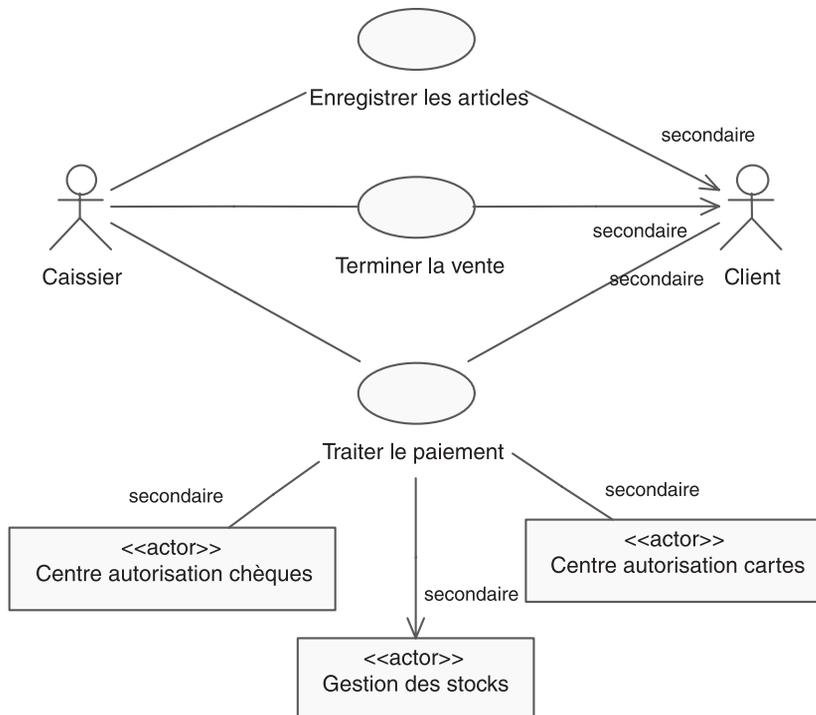
À retenir ACTEUR RÉCEPTEUR UNIQUEMENT

Notez l'utilisation de la flèche de navigabilité sur l'association avec l'acteur non-humain *Gestion des stocks* qui permet de préciser que l'acteur ne fait que recevoir des messages du système, sans jamais lui en envoyer.

Cette prolifération d'acteurs secondaires nous amène à diagnostiquer que ce cas d'utilisation a trop de responsabilités, et qu'il est souhaitable de le découper en parties plus atomiques. On pourrait penser qu'il suffit de le scinder séquentiellement comme cela est illustré sur la figure 2-3.

Figure 2-3.

Découpe séquentielle du cas d'utilisation principal

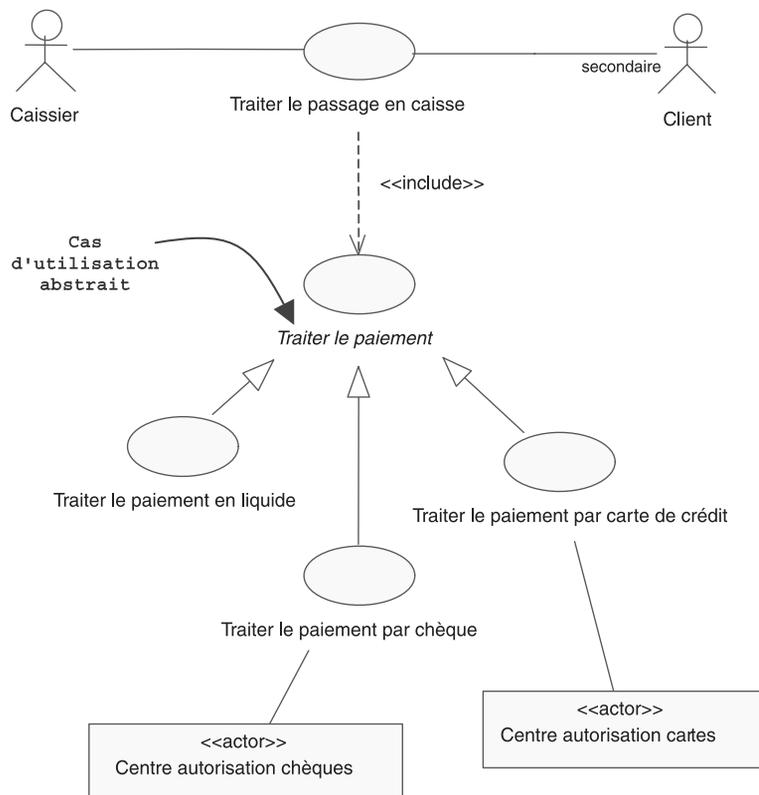


Quoique tentante, cette solution est rarement recommandable. En effet, les cas d'utilisation qui en résultent ne répondent plus vraiment à la définition UML. Peut-on par exemple considérer que *Terminer la vente* représente un service de bout en bout qui est rendu par le système ?

Le niveau de détail ainsi obtenu s'apparente plutôt à ce que Larman appelle des *opérations système*², soit à une unité de traitement qui est réalisée par le système dans le cadre d'un cas d'utilisation et qui peut éventuellement être réutilisée dans un autre.

L'enregistrement des articles et la clôture de la vente font intervenir les mêmes acteurs et se suivent forcément dans le temps : il n'y a donc pas de raison de les séparer. En revanche, l'importante partie variable, qui est liée au choix du mode de paiement par le client, conduit à séparer, grâce à une relation d'inclusion, la procédure générique de paiement du processus englobant de traitement du passage en caisse. Cela permet ainsi de décrire des cas d'utilisation spécialisés, faisant chacun apparaître des acteurs spécifiques. Le début de l'énoncé peut donc se modéliser comme cela est représenté sur la figure suivante.

Figure 2-4.
Diagramme de cas
d'utilisation partiel



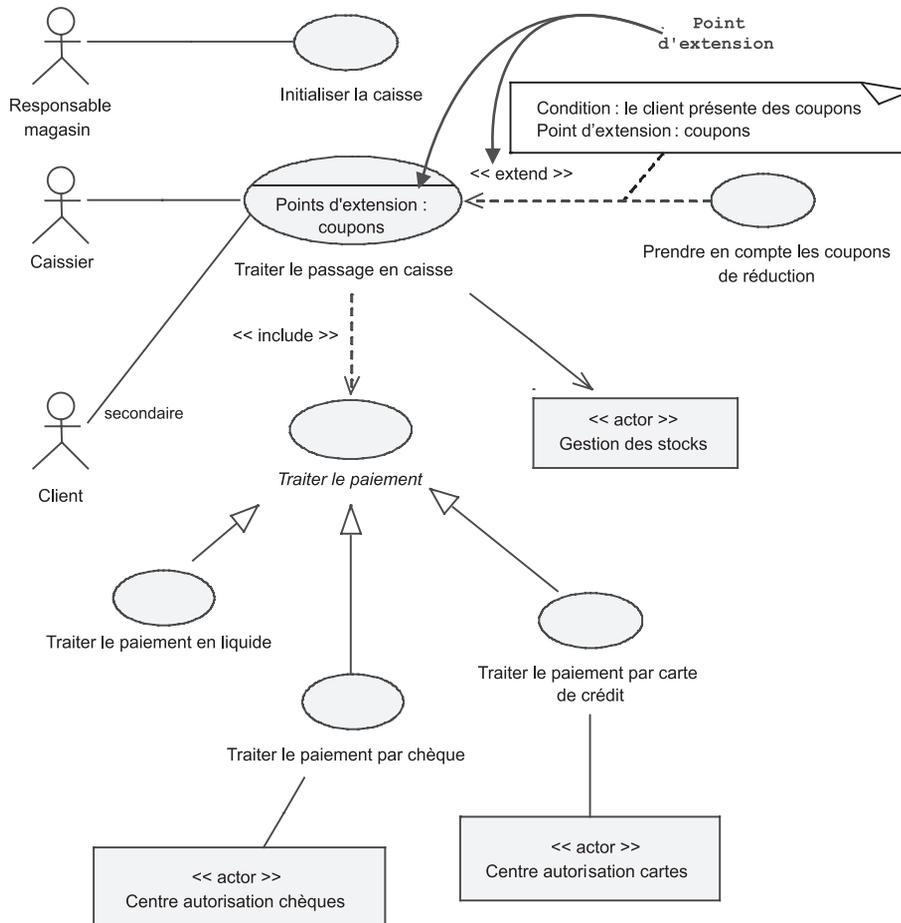
2. Par analogie avec les opérations que sont capables de réaliser les objets, suite à la réception de messages venant d'un autre objet. À ce niveau, le système boîte noire est vu comme un « gros » objet, et les acteurs lui envoient des messages qui déclenchent des opérations « système ».

Le cas d'utilisation inclus, *Traiter le paiement*, est porté en italique sur le diagramme pour indiquer qu'il s'agit d'un cas d'utilisation abstrait (non instanciable). Pour ne pas surcharger le schéma, nous n'avons pas reporté les associations avec *Caissier* et *Client* sur *Traiter le paiement*. On notera cependant que deux cas d'utilisation spécialisés possèdent une association spécifique avec un acteur supplémentaire : le centre d'autorisation les concernant.

Nous pouvons maintenant compléter le modèle en intégrant la fin de l'énoncé.

La prise en compte optionnelle des bons de réduction se traduit assez naturellement par une relation d'extension avec le cas d'utilisation principal. La liaison avec le système externe de gestion des stocks donne lieu à une association unidirectionnelle avec un nouvel acteur. L'initialisation de la caisse ne présente pas de difficulté. Le diagramme de cas d'utilisation complété est présenté ci-après.

Figure 2-5.
Diagramme de cas d'utilisation complété



Étape 2 – Descriptions essentielle et réelle d'un cas d'utilisation



EXERCICE 2-2.

Description essentielle d'un cas d'utilisation

À retenir

CAS D'UTILISATION ESSENTIEL/RÉEL

C. Larman a introduit dans [Larman 97] la distinction entre cas d'utilisation essentiel et cas d'utilisation réel :

- *Essentiel* : décrit un processus, d'un point de vue analytique. Explicite un processus le plus indépendamment possible de l'environnement matériel/logiciel.
- *Réel* : décrit un processus, du point de vue de la conception. Explicite une solution en termes d'événements, d'interface utilisateur, d'entrées de données, etc.

Nous allons illustrer cette différence avec les deux questions suivantes.

Écrivez une description détaillée essentielle du cas d'utilisation principal :
TRAITER LE PASSAGE EN CAISSE.

solution

Sommaire d'identification

Titre : Traiter le passage en caisse **Type** : essentiel détaillé

Résumé : un client arrive à une caisse avec des articles qu'il souhaite acheter. Le caissier enregistre les achats et récupère le paiement. À la fin de l'opération, le client part avec les articles.

Acteurs : Caissier (principal), *Client (secondaire)*.

Date de création : 17/10/03

Date de mise à jour : 11/05/06

Version : 1.5

Responsable : Pascal Roques

Description des scénarios

Préconditions

Le TPV est en service ; un caissier y est connecté ;

Le catalogue produit est disponible.

Scénario nominal

1. Ce cas d'utilisation commence quand un client arrive à la caisse avec des articles qu'il souhaite acheter.	
2. Le Caissier enregistre chaque article. S'il y a plus d'un exemplaire par article, le Caissier indique également la quantité.	3. Le TPV valide le CPU et détermine le prix de l'article. Le TPV affiche la description et le prix de l'article en question.
4. Après avoir enregistré tous les articles, le Caissier indique que la vente est terminée.	5. Le TPV calcule et affiche le montant total de la vente.
6. Le Caissier annonce le montant total au client.	
7. Le Client choisit le type de paiement : a. En cas de paiement cash, exécuter le cas d'utilisation « Traiter le paiement en liquide ». b. En cas de paiement par carte de crédit, exécuter le cas d'utilisation « Traiter le paiement par carte de crédit ». c. En cas de paiement par chèque, exécuter le cas d'utilisation « Traiter le paiement par chèque ».	
	8. Le TPV enregistre la vente effectuée et imprime un ticket.
9. Le Caissier donne le ticket de caisse au Client.	
10. Le Client s'en va avec les articles qu'il a achetés.	

Enchaînements alternatifs

A1 : numéro d'identification inconnu

L'enchaînement A1 démarre au point 3 du scénario nominal.

3. Le TPV indique au Caissier que le numéro d'identification de l'article est inconnu. L'article ne peut alors être pris en compte dans la vente en cours.

Le scénario nominal reprend au point 2 s'il y a d'autres articles ou au point 4 s'il n'y en a pas.

A2 : demande d'annulation d'article

L'enchaînement A2 démarre au point 2 du scénario nominal.

2. Le Caissier demande l'annulation du dernier article (prix erroné, etc.).

3. Le TPV enlève l'article de la vente en cours.

Le scénario nominal reprend au point 2 s'il y a d'autres articles ou au point 4 s'il n'y en a pas.

Enchaînements d'erreur

E1 : annulation de la vente

L'enchaînement E1 peut démarrer du point 2 au point 7 du scénario nominal.

2.7 Le Caissier annule l'ensemble de la vente et le cas d'utilisation se termine en échec.

Postconditions

- La vente est enregistrée dans le TPV.

Il faudrait également décrire chacun des cas d'utilisation spécialisés.

Nous ne corrigeons que le premier :

Sommaire d'identification

Titre : Traiter le paiement en liquide

Résumé : un client paie en liquide le total affiché par la caisse enregistreuse.

Acteurs : Caissier (principal), *Client (secondaire)*.

Date de création : 17/10/03

Date de mise à jour : 11/05/06

Version : 1.5

Responsable : Pascal Roques

Description des scénarios

Préconditions

La saisie des articles de la vente en cours est clôturée.

Scénario nominal

1. Ce cas d'utilisation commence quand un Client choisit de payer en espèces, après avoir été informé du montant total de la vente.	
2. Le Client donne en paiement une somme en espèces ; elle est éventuellement plus élevée que le montant total de la vente.	
3. Le Caissier enregistre la somme donnée par le client.	4. Le TPV affiche la somme qui doit être rendue au Client.
5. Le Caissier encaisse l'argent reçu et sort la monnaie qu'il doit rendre.	
6. Le Caissier rend la monnaie au Client.	

Enchaînements alternatifs ou d'erreur

A1 : somme donnée par le client inférieure au montant total de la vente

L'enchaînement A1 démarre au point 4 du scénario nominal.

2. Le TPV signale que la somme donnée par le Client est inférieure au montant total de la vente et invite le Caissier à recommencer.

Le scénario nominal reprend au point 3.

E1 : client ne pouvant payer

L'enchaînement E1 démarre au point 2 du scénario nominal.

2. Le client n'a pas assez de liquide pour payer.
3. Le caissier annule l'ensemble de la vente et le cas d'utilisation se termine en échec, ou le client paie avec un autre moyen de paiement (Voir « Traiter le paiement par chèque », ou « Traiter le paiement par carte de crédit »).

E2 : caissier ne pouvant rendre la monnaie

L'enchaînement E2 démarre au point 5 du scénario nominal.

5. Le tiroir du TPV ne contient pas assez d'espèces pour qu'il soit possible de rendre la monnaie.
6. Le caissier demande des espèces supplémentaires à son supérieur ou propose une autre méthode de paiement au client (Voir « Traiter le paiement par chèque », ou « Traiter le paiement par carte de crédit »). Si aucune solution n'est envisageable, le caissier annule l'ensemble de la vente et le cas d'utilisation se termine en échec.



EXERCICE 2-3.

Description réelle d'un cas d'utilisation

Écrivez une description détaillée réelle du cas d'utilisation principal : TRAITER LE PASSAGE EN CAISSE.

Proposez tout d'abord une fenêtre graphique simple pour l'interface homme-machine du caissier.

solution

Le sommaire d'identification est similaire au précédent, mais le type devient : réel détaillé.

L'interface homme-machine proposée est la suivante :

La description du scénario nominal devient alors :

1. Ce cas d'utilisation commence quand un Client arrive à la caisse avec des articles qu'il souhaite acheter.	
2. Le Caissier enregistre le code universel d'identification du produit dans le champ « Numéro » de la fenêtre de dialogue de la caisse enregistreuse. S'il y a plus d'un exemplaire de l'article en question, le Caissier peut entrer la quantité dans le champ « Quantité », qui est positionné à « 1 » par défaut. Puis il appuie sur le bouton de validation : « Saisie article ».	3. Le TPV détermine le prix de l'article et ajoute les informations sur l'article à la vente en cours. Le TPV affiche la description (sur 6 lettres) et le prix de l'article en question dans le champ « Total ».
4. Après avoir enregistré tous les articles, le Caissier appuie sur le bouton « Fin de vente ».	5. Le TPV calcule et affiche le montant total de la vente dans le champ « Total ».
6. Le Caissier annonce le montant total au Client.	
7. Le Client choisit le type de paiement : <ul style="list-style-type: none"> • En cas de paiement cash, exécuter le cas d'utilisation « Traiter le paiement en liquide ». • En cas de paiement par carte de crédit, exécuter le cas d'utilisation « Traiter le paiement par carte de crédit ». • En cas de paiement par chèque, exécuter le cas d'utilisation « Traiter le paiement par chèque ». 	

	8. Le TPV enregistre la vente qui vient d'être effectuée et imprime un ticket.
9. Le Caissier donne le ticket de caisse au Client.	
10. Le Client s'en va avec les articles qu'il a achetés.	

Pour compléter, nous donnons ci-après la version réelle de *Traiter le paiement en liquide*.

Scénario nominal

1. Ce cas d'utilisation commence quand un Client choisit de payer en espèces, après avoir été informé du montant total de la vente.	
2. Le Client donne en paiement une somme en espèces ; elle est éventuellement plus élevée que le montant total de la vente.	
3. Le Caissier enregistre la somme donnée par le client dans le champ « Paiement ». Il valide au moyen du bouton « Saisie paiement ».	4. Le TPV affiche la somme qui doit être rendue au Client dans le champ « À rendre ».
5. Le Caissier encaisse l'argent reçu et sort la monnaie qu'il doit rendre.	
6. Le Caissier rend la monnaie au client.	

Étape 3 – Description graphique des cas d'utilisation



EXERCICE 2-4.

Diagramme de séquence système

Réalisez un diagramme de séquence système qui décrive le scénario nominal du cas d'utilisation essentiel TRAITER LE PASSAGE EN CAISSE, en ne considérant que le paiement cash.

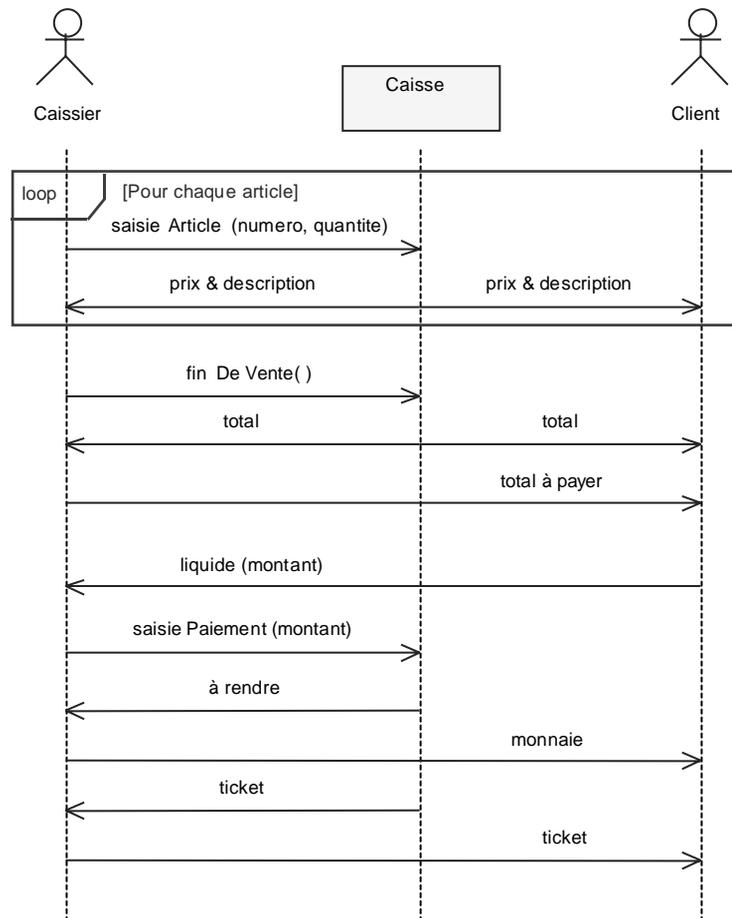
solution

Il suffit de transcrire sous forme de diagramme de séquence les interactions citées dans le scénario textuel de la réponse 2-2, en utilisant les conventions graphiques adoptées précédemment :

- l'acteur principal *Caissier* à gauche ;
- un participant représentant le TPV au milieu ;
- l'acteur secondaire *Client* à droite de la Caisse.

Notez également l'utilisation du fragment d'interaction proposée par UML 2, avec l'opérateur de boucle *loop*.

Figure 2-6.
Diagramme de séquence
système du scénario
nominal de Traiter le
passage en caisse



Deux commentaires sur le diagramme précédent :

- Nous avons choisi de montrer les messages échangés par les acteurs. Cela n'est pas strictement nécessaire, puisqu'en dehors du périmètre du système, mais peut néanmoins être utile au lecteur afin de valider le diagramme. Nous avons

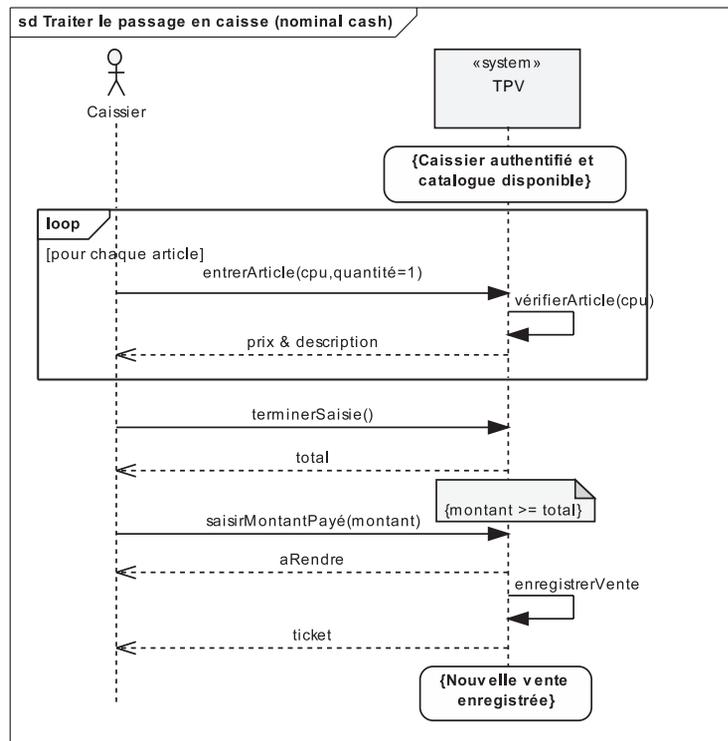
effectivement plus représenté un processus métier qu'un strict cas d'utilisation, mais cela est souvent apprécié par l'expert métier. Comparez avec la figure suivante qui se concentre vraiment sur les interactions avec le système.

- Comme le prix et la description sont envoyés simultanément aux deux acteurs, nous avons simplement dessiné deux flèches partant du même niveau vertical. Cela nous a paru plus clair pour le lecteur que l'utilisation d'un autre opérateur UML 2, à savoir « seq » (*weak sequencing*).

Si l'on considère que l'ajout de l'acteur Client sur la figure précédente n'apporte pas de réelle plus-value, on peut simplifier le diagramme de séquence en enlevant sa ligne de vie.

Par contre, si l'on utilise un formalisme un peu plus sophistiqué, avec le cadre de diagramme, les flèches de retour pointillées, les messages à soi-même pour les traitements du système, et la description des pré- et post-conditions grâce au symbole d'état sur la ligne de vie, on obtient finalement le diagramme suivant :

Figure 2-7.
Version plus élaborée du DSS
du scénario nominal
de Traiter le passage en caisse



On peut maintenant essayer de compléter le diagramme en indiquant les scénarios alternatifs et d'erreur grâce à des notes (voir figure 2-8).

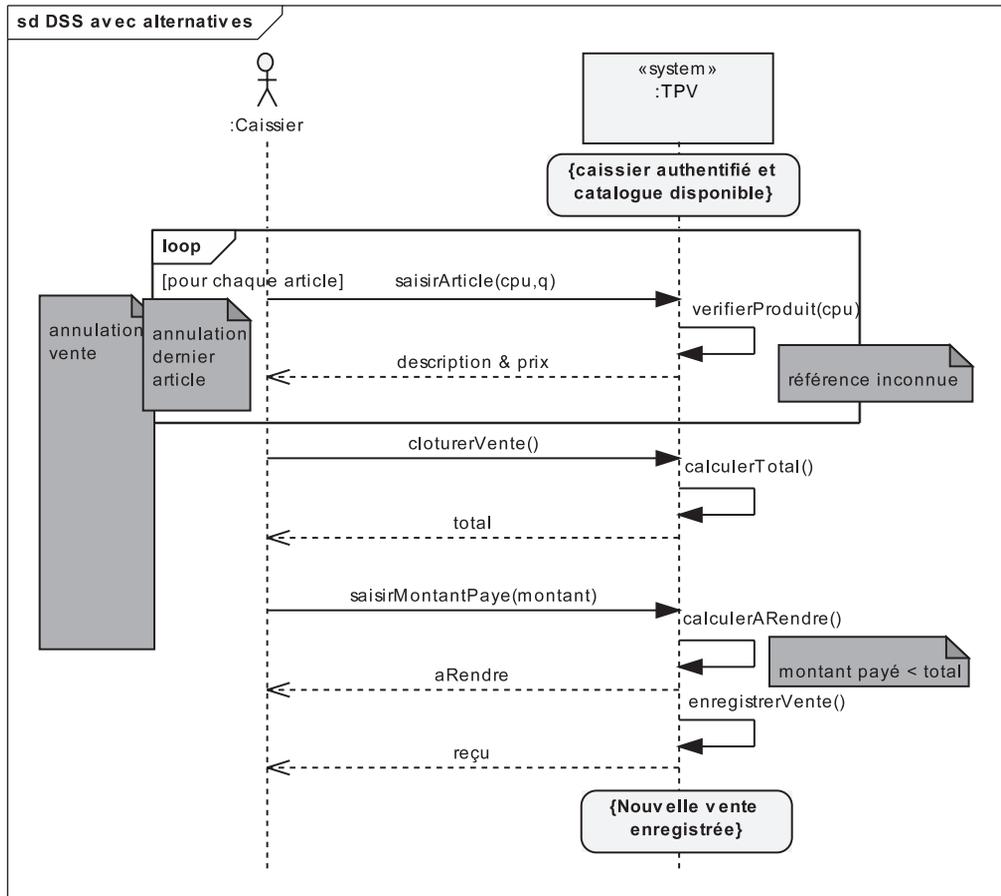


Figure 2-8.

DSS de Traiter le passage en caisse avec alternatives et erreurs



EXERCICE 2-5.

Diagramme de séquence système (suite)

Modifiez le diagramme de séquence précédent afin de prendre en compte les différents types de paiement. Proposez également un diagramme supplémentaire montrant le travail du caissier pendant les heures d'ouverture du magasin.

solution

Pour prendre en compte les différents types de paiement, on peut remplacer le message `saisirMontantPayé(montant)` du diagramme précédent par une référence vers une interaction concernant le cas d'utilisation inclus *Traiter le paiement*.

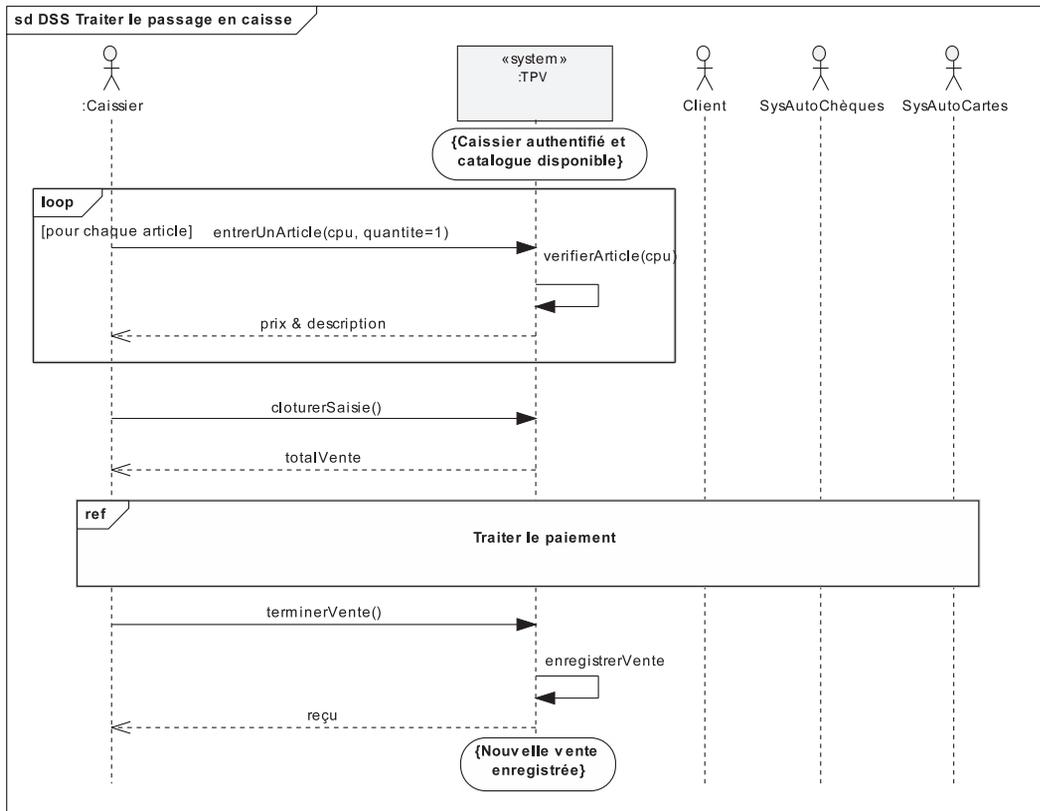


Figure 2-9.

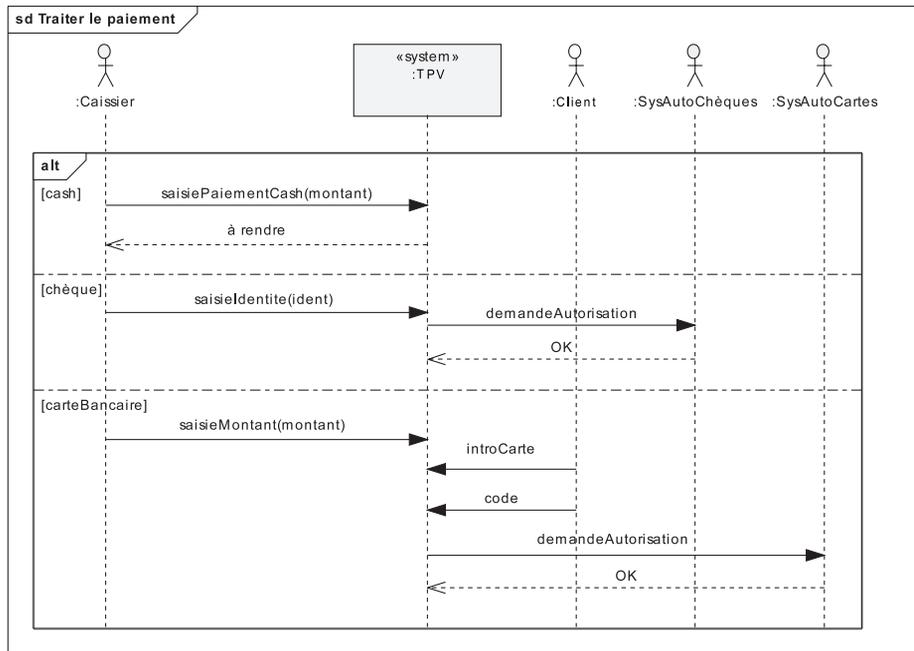
Version étendue du DSS

du scénario nominal de Traiter le passage en caisse

Notez que nous devons néanmoins faire figurer les acteurs secondaires pour assurer la cohérence des lignes de vie entre les diagrammes 2-9 et 2-10.

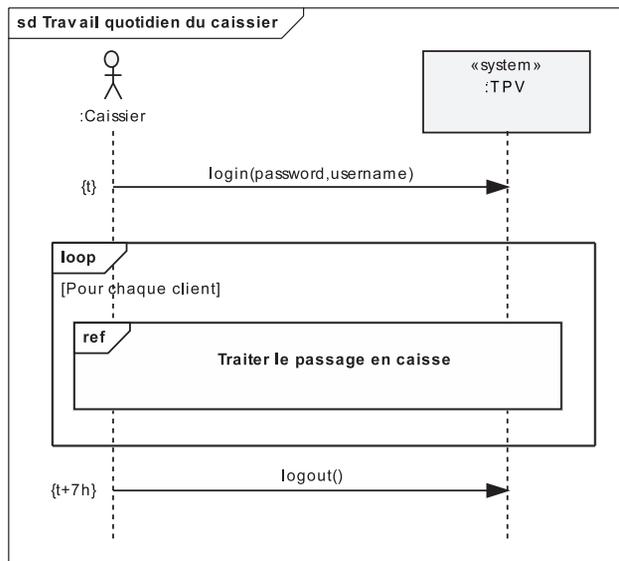
Le diagramme de séquence décrivant le cas d'utilisation *Traiter le paiement* est donné ci-après. Il utilise un fragment conditionnel (opérateur *alt*).

Figure 2-10.
DSS
conditionnel
du scénario
nominal
de Traiter le
paiement



En ce qui concerne le travail quotidien du caissier, il consiste à traiter le passage en caisse des clients successifs, après s’être authentifié en arrivant au magasin et avant de sortir de l’application en partant. Si nous voulons ajouter une contrainte de temps relative aux 35 heures, soit une moyenne de 7 heures de travail consécutif par jour, nous obtenons le diagramme de séquence suivant.

Figure 2-11.
Diagramme de séquence du travail
quotidien du caissier



Étape 4 – Réalisation d'un diagramme d'états au niveau système



EXERCICE 2-6.

Diagramme d'états des opérations système

Montrez par un diagramme d'états la succession forcée des opérations système pour le cas d'utilisation TRAITER LE PASSAGE EN CAISSE, en ne considérant toujours que le paiement cash. Étendez ensuite le diagramme en prenant en compte les différents types de paiement, ainsi que les autres actions du caissier.

solution

Les opérations système identifiées grâce à l'exercice 2-4 correspondent aux trois messages³ dont le système est destinataire :

Figure 2-12.

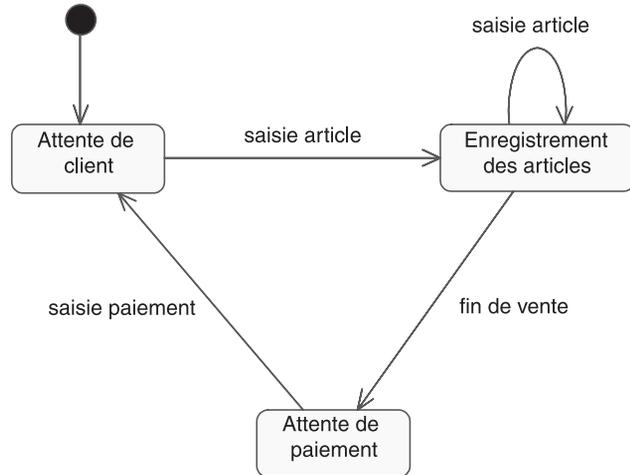
Opérations système de
Traiter le passage en caisse

<<system>> Caisse
saisieArticle(numero, quantite) finDeVente() saisiePaiement(montant)

Pour représenter la séquence forcée de ces trois opérations système, avec la répétition possible de la saisie d'article, un diagramme d'états s'impose⁴. Il représente en fait le sous-ensemble des états de la caisse induits par le cas d'utilisation *Traiter le passage en caisse*.

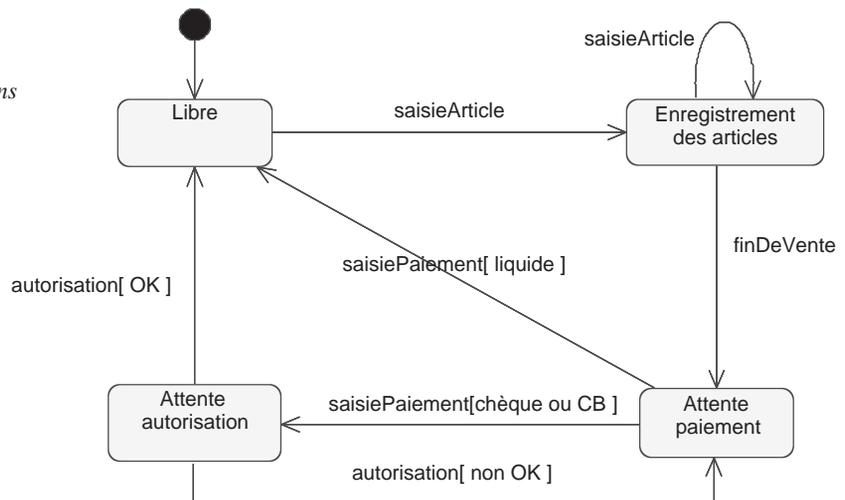
3. Pour simplifier, nous n'avons pas pris en compte les deux messages d'annulation : annulerArticle et annulerVente. La démarche de prise en compte de ces deux messages serait identique : ajouter les transitions correspondantes sur le diagramme d'états 2.13.
4. UML 2 parle dans ce cas de machine à états de type protocole (*protocol state machine*). Contrairement à une machine à états de type comportement qui décrit les réactions d'un objet en réponse à des événements, une machine à états de type protocole spécifie les séquences légales des événements qui peuvent se produire dans le contexte d'une classe ou d'un composant.

Figure 2-13.
Diagramme d'états des opérations système de Traiter le passage en caisse



Si l'on tient compte des différents types de paiement, le diagramme d'états devient légèrement plus complexe.

Figure 2-14.
Diagramme d'états complété des opérations système de Traiter le passage en caisse

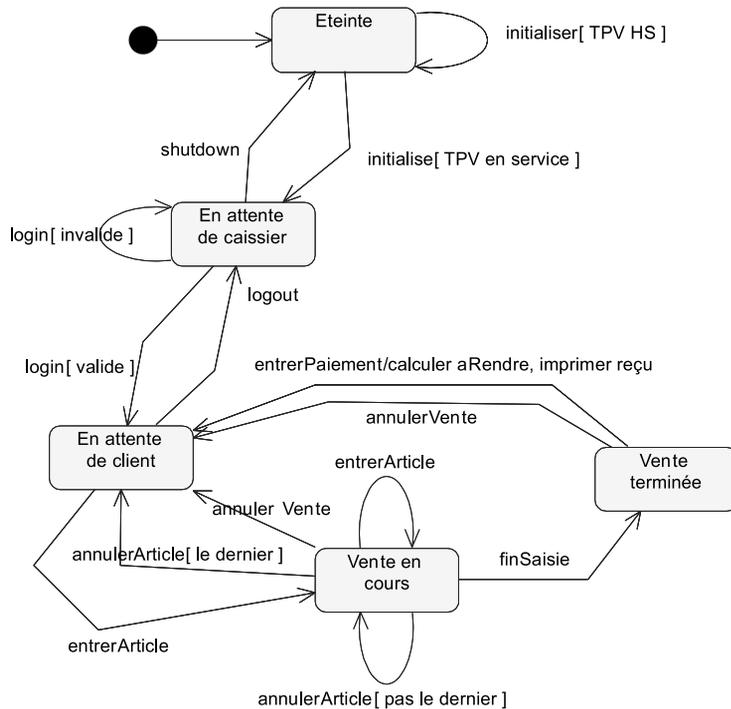


Enfin, pour compléter le diagramme d'états de la caisse, indépendamment d'un cas d'utilisation particulier, il faudrait ajouter des états supplémentaires liés par exemple à l'initialisation de la caisse, à la connexion du caissier, etc.

Voici un exemple plus complet de ce que l'on pourrait alors obtenir, en gardant la restriction du paiement cash (donc à comparer plutôt à la figure 2-13) :

Figure 2-15.

Diagramme d'états de la caisse (paiement cash)



CONSEILS MÉTHODOLOGIQUES

COMMENT IDENTIFIER LES ACTEURS ?

- Les acteurs sont *a priori* :
 - les utilisateurs humains directs : identifiez tous les profils possibles, sans oublier l'administrateur, l'opérateur de maintenance, etc. ;
 - les autres systèmes connexes qui interagissent aussi directement avec le système étudié, souvent par le biais de protocoles bidirectionnels.
- Éliminez autant que faire se peut les acteurs « physiques » au profit des acteurs « logiques » : l'acteur est celui qui bénéficie de l'utilisation du système. Cette règle permet en particulier de s'affranchir des technologies d'interface qui risquent d'évoluer au fil du projet et d'identifier des acteurs susceptibles d'être mieux réutilisés.
- Répertoirez en tant qu'acteurs uniquement les entités externes (et pas des composants internes au système étudié) qui interagissent directement avec le système (et pas indirectement par le biais d'autres acteurs).

- Ne confondez pas rôle et entité concrète. Une même entité externe concrète peut jouer successivement différents rôles par rapport au système étudié, et être modélisée par plusieurs acteurs. Réciproquement, le même rôle peut être tenu simultanément par plusieurs entités externes concrètes, qui seront alors modélisées par le même acteur.

COMMENT REPRÉSENTER LES ACTEURS ?

Utilisez la forme graphique du *stick man* pour les acteurs humains, et la représentation rectangulaire avec le mot-clé <<actor>> pour les systèmes connectés.

LE DIAGRAMME DE CONTEXTE STATIQUE

Réalisez si nécessaire un *diagramme de contexte statique*. Il suffit pour cela d'utiliser un diagramme de classes dans lequel chaque acteur est relié par une association à une classe centrale unique représentant le système, ce qui permet de spécifier le nombre d'instances d'acteurs connectés au système à un moment donné.

COMMENT IDENTIFIER LES CAS D'UTILISATION ?

Utilisez la procédure systématique suivante :

- délimitez le sujet : les frontières du système à l'étude, son périmètre. S'agit-il d'une application logicielle, d'un système intégrant du matériel et du logiciel, de cet ensemble plus son opérateur ou l'organisation entière qui l'opère ?
- identifiez les acteurs principaux, c'est-à-dire ceux qui cherchent à obtenir un résultat observable, un but, en utilisant les services du système ;
- identifiez les objectifs, les buts, de chaque acteur principal ;
- définissez des cas d'utilisation autonomes qui correspondent aux objectifs des acteurs principaux.

Ne vous inquiétez pas : il est normal que tous les objectifs et les cas d'utilisation ne soient pas correctement identifiés d'emblée. Il s'agit forcément d'un processus de recherche itératif et évolutif.

COMMENT AMÉLIORER LE DIAGRAMME DE CAS D'UTILISATION ?

Pour améliorer le contenu informatif du diagramme de cas d'utilisation, nous recommandons d'adopter les conventions suivantes :

- par défaut, le rôle d'un acteur est principal ; si ce n'est pas le cas, indiquez explicitement que le rôle est secondaire sur l'association, du côté de l'acteur ;
- dans la mesure du possible, disposez les acteurs principaux à gauche des cas d'utilisation, et les acteurs secondaires à droite ;

- si l'acteur ne fait que recevoir des messages du système (ou seulement en envoyer), utilisez le symbole de restriction de navigabilité sur l'association entre le cas d'utilisation et l'acteur.

COMMENT DÉCRIRE LES CAS D'UTILISATION ?

- Pour détailler la dynamique du cas d'utilisation, la façon la plus simple de procéder consiste à recenser textuellement toutes les interactions entre les acteurs et le système. Le cas d'utilisation doit avoir un début et une fin clairement identifiés. Il faut également préciser les variantes possibles, telles que le scénario nominal, les différents cas « alternatifs », les cas d'erreurs, tout en essayant d'ordonner séquentiellement les descriptions, afin d'améliorer leur lisibilité.
- Ne mélangez pas cas d'utilisation *essentiel* (indépendant de tout choix technologique d'interface avec les acteurs) et cas d'utilisation *réel* : le premier est beaucoup plus stable et peut être plus facilement réutilisé.
- Rédigez avec concision en évitant les mots inutiles. Généralisez l'adoption de conventions communes au sein du projet. Celles proposées par A. Cockburn dans son ouvrage *Rédiger des cas d'utilisation efficaces* [Cockburn 01] ont le mérite d'être largement répandues et acceptées.
- Rédigez des cas d'utilisation en « boîte noire » : ne décrivez pas le fonctionnement interne du système, ses composants ou sa conception. Considérez uniquement ce qui est visible par les acteurs.
- Complétez la description textuelle des cas d'utilisation par un ou plusieurs diagrammes dynamiques UML :
 - Pour la dynamique globale du cas d'utilisation, utilisez un *diagramme d'activité* ou un *diagramme de séquence* en profitant des nouvelles constructions UML 2 : fragments avec opérateurs, références entre interactions. Vous pouvez également considérer le nouveau diagramme appelé *Interaction Overview Diagram*, qui est une fusion du diagramme d'activité et du diagramme de séquence.
 - pour décrire le scénario nominal, utilisez un *diagramme de séquence*. Présentez-le en montrant l'acteur principal à gauche, puis un objet qui représente le système en boîte noire, et enfin en disposant à droite du système les éventuels acteurs secondaires sollicités durant le scénario.
- Vous pouvez enrichir le diagramme de séquence système du scénario nominal de façon à faire apparaître également :
 - les principales actions internes du système (au moyen de messages qu'il s'envoie à lui-même) ;
 - les renvois aux enchaînements alternatifs et d'erreur (au moyen de notes) ;
 - les parties répétées, optionnelles ou alternatives (opérateurs *loop*, *opt* et *alt*).

LES RELATIONS ENTRE CAS D'UTILISATION

- Utilisez la relation d'inclusion entre cas d'utilisation pour éviter de devoir décrire plusieurs fois le même enchaînement, en factorisant ce comportement commun dans un cas d'utilisation supplémentaire inclus.
- Utilisez la relation d'extension entre cas d'utilisation pour séparer un comportement complexe optionnel ou rare du comportement obligatoire.
- Utilisez la relation de généralisation/spécialisation entre cas d'utilisation pour formaliser des variations importantes sur le même cas d'utilisation.
- N'abusez cependant pas des relations entre cas d'utilisation (surtout extension et généralisation) : elles peuvent rendre les diagrammes de cas d'utilisation difficiles à décrypter pour les experts métier qui sont censés les valider.

PAS PLUS DE 20 CAS D'UTILISATION !

Limitez à 20 le nombre de vos cas d'utilisation de base (en dehors des cas inclus, spécialisés, ou des extensions). Avec cette limite arbitraire, on reste synthétique et on ne tombe pas dans le piège de la granularité trop fine des cas d'utilisation.

SOYEZ AGILES ET ITÉRATIFS !

Les descriptions textuelles des cas d'utilisation et les diagrammes UML ne sont jamais parfaits. Il leur manque certains éléments et certaines assertions sont fausses. La solution ne consiste pas à adopter l'attitude du processus en cascade et à déployer toujours plus d'efforts pour obtenir d'emblée des spécifications exactes et exhaustives. Il s'agit simplement de faire de notre mieux dans les délais impartis et en appliquant les meilleures pratiques.

Il existe un moyen terme entre la démarche en cascade et la programmation sauvage : le développement itératif et incrémental. Dans cette approche, les cas d'utilisation et les autres modèles sont progressivement affinés, vérifiés et clarifiés grâce à la programmation et aux tests.

