**Pascal Roques** Préface de Gaël Renault

Conception et programmation orientées objet 8e éditionUML 2.5 par la pratique Études de cas et exercices corrigés

**Enfin un ouvrage détaillant tous les principes d’une modélisation efficace avec UML ! En mettant l’accent sur les diagrammes les plus importants (cas d’utilisation, séquence, classes, états, activité), l’auteur se concentre sur les principaux concepts et fournit des explications claires et pragmatiques applicables à vos projets. De nombreux conseils émaillent les multiples exercices proposés afin de faciliter l’assimilation du langage UML. Huitième édition augmentée : un cours pratique magistral sur UML 2.5**

**Cette huitième édition mise à jour et augmentée de l’ouvrage *UML par la pratique* constitue un support de cours exemplaire sur UML 2.5. Il traite les axes fonctionnel, statique et dynamique de la modélisation UML par des études de cas et des exercices corrigés donnant les bases d’une démarche méthodique.**

**Chaque choix de modélisation est minutieusement commenté ; des conseils issus de l’expérience de l’auteur ainsi que de nombreux avis d’experts sont donnés. En fin d’ouvrage, un glossaire reprend les définitions des principaux concepts étudiés.**

**Les nouveaux concepts et diagrammes UML 2 sont traités en détail : diagramme de structure composite, nouveautés du diagramme d’activité et du diagramme de séquence, etc., en tenant compte des méthodes de développement agiles. Enfin, une étude de cas complète illustre le processus de développement itératif depuis la modélisation métier jusqu’à la conception détaillée en Java et C#. À qui s’adresse ce livre ?**

• **Aux étudiants en informatique (cursus génie logiciel ou modélisation UML) et à leurs professeurs, qui y trouveront un matériel précieux pour illustrer par des cas réels les concepts étudiés en cours.**

• **À toutes les personnes impliquées dans des projets de développement logiciel : maîtres d’ouvrage, chefs de projet, analystes et concepteurs, architectes logiciel, développeurs, etc.**

**Au sommaire Point de vue fonctionnel.** Qualité logicielle • Modélisation fonctionnelle : étude de cas • Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques • **Point de vue statique.** Modélisation statique : étude de cas • Modélisation statique : exercices corrigés et conseils méthodologiques • **Point de vue dynamique.** Modélisation dynamique : étude de cas • Modélisation dynamique : exercices corrigés et conseils méthodologiques • **Conception.** Étude de cas complète : de la modélisation métier à la conception détaillée en Java ou C# • Études de cas complémentaires • Conclusion • **Annexes**

Consultant senior et formateur indépendant, **Pascal Roques** a plus de vingt-cinq ans d’expérience dans la modélisation de systèmes complexes (SADT, OMT, UML, SysML). Il a ainsi été responsable des formations Valtech Training sur le thème « Analyse, conception et modélisation avec UML » pendant de nombreuses années. Auteur de plusieurs livres consacrés à UML et SysML aux éditions Eyrolles, il a obtenu la certification OMG-Certified UML Advanced Professional proposée par l’OMG.

**P**

Conception programma orientées o UML 2.5 par la prat Études et exe

Récapitulatif (simplifié) de la démarche de modélisation

Cas d'utilisation

**Enplu exp deHuitième é**

**CeUMdoChavLed’a coÀ qui s’adr**

Diagramme de séquence "système"

Diagramme de séquence de conception

Diagramme de classes d'analyse

Diagramme de classes de conception

**ApÀe**

Cocommo il a**Au som P**etcoetde

Cocommo il a**Au som P**etcoetde

UML 2.5 par la pratique

296723AWP-UML2\_8E.book Page 2 Lundi, 19. février 2018 7:54 07

**DANS LA MÊME COLLECTION**

H. BERSINI, P. ALEXIS, G. DEGOLS. – **Apprendre la programmation web avec Python et Django.** N° 67515, 2e édition, 2018, 396 pages.

H. BERSINI. – **La programmation orientée objet.** N° 67399, 7e édition, 2017, 696 pages.

C. DELANNOY. – **Programmer en langage C++.** N° 67386, 9e édition, 2017, 850 pages.

C. SOUTOU. – **Programmer avec MySQL.** N° 67379, 5e édition, 2017, 522 pages.

C. DELANNOY. – **Exercices en langage C++.** N° 67387, 3e édition, 2017, 354 pages.

C. DELANNOY. – **S’initier à la programmation et à l’orienté objet.** *Avec des exemples en C, C++, C#, Python, Java et PHP.* N° 14011, 2e édition, 2016, 360 pages.

A. TASSO. – **Le livre de Java premier langage.** N° 14384, 11e édition, 2016, 598 pages.

C. DELANNOY. – **Programmer en Java.** N° 11889, 9e édition, 2016, 948 pages.

E. SARRION. – **Programmation avec Node.js, Express.js et MongoDB.** N° 13994, 2014, 586 pages.

**SUR LE MÊME THÈME**

P. ROQUES. – **Mémento UML 2.5.** N° 14356, 2015, 14 pages.

P. ROQUES. – **UML 2 : modéliser une application web.** N° 12389, 4e édition, 2008, 246 pages.

X. BLANC, I. MOUNIER. – **UML 2 pour les développeurs.** N° 12029, 2006, 202 pages.

P. ROQUES, F. VALLÉE. – **UML 2 en action.** N° 11213, 2e édition, 2002, 400 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur **http://izibook.eyrolles.com**

**Pascal Roques** Préface de Gaël Renault 8e éditionUML 2.5 par la pratique Études de cas et exercices corrigés

ÉDITIONS EYROLLES 61, bd Saint-Germain 75240 Paris Cedex 05 www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l’autorisation de l’Éditeur ou du Centre français d’exploitation du droit de copie, 20, rue des Grands-Augustins, 75006 Paris.

© Groupe Eyrolles, 2001, 2002, 2004, 2005, 2006, 2008, 2009, 2018, ISBN : 978-2-212-67565-8

296723AWP-UML2\_8E.book Page 5 Lundi, 19. février 2018 7:54 07

*A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B.*

Douglas. T. Ross

*La différence entre la théorie et la pratique, c’est qu’en théorie, il n’y a pas de différence entre la théorie et la pratique, mais qu’en pratique, il y en a une.*

Jan van de Sneptscheut

*Depuis des temps très anciens, les hommes ont cherché un langage à la fois universel et synthétique, et leurs recherches les ont amenés à découvrir des images, des symboles qui expriment, en les réduisant à l’essentiel, les réalités les plus riches et les plus complexes.*

*Les images, les symboles parlent, ils ont un langage.*

O.M. Aïvanhov

296723AWP-UML2\_8E.book Page 6 Lundi, 19. février 2018 7:54 07

296723AWP-UML2\_8E.book Page 7 Lundi, 19. février 2018 7:54 07

Préface

Adopté et standardisé par l’Object Management Group depuis 1997, UML est aujourd’hui un outil de communication incontournable, utilisé sur des centaines de projets de par le monde ; en conséquence, la connaissance d’UML est désormais une des compétences qui sont exigées quasi systématiquement lors d’un recrute- ment. Pourtant, trop nombreux sont encore les concepteurs qui s’imaginent, à tort, posséder cette compétence parce qu’ils connaissent la représentation graphique d’une classe d’association, d’une activité ou d’un état, tandis qu’ils ne savent pas expliquer ni défendre leur emploi dans un modèle, même simple. C’est que, malgré la rigueur apportée à la spécification d’UML, il y a parfois différentes façons de représenter une même idée, ou, à l’inverse, une idée donnée pourra être représentée avec plus ou moins de précision selon que l’on aura utilisé telle ou telle particularité syntaxique d’UML.

Si l’on trouve aujourd’hui pléthore de formations à UML, il faut bien reconnaître que la plupart d’entre elles se trompent d’objectif. L’enseignement d’UML en tant que tel ne présente pas plus de difficulté ni d’intérêt que l’enseignement de n’importe quel autre langage de modélisation. Toutefois, ce qui importe en la matière, c’est d’ensei- gner efficacement les principes qui sous-tendent une démarche objet dédiée à l’analyse et à la conception de systèmes complexes, mais aussi qu’un emploi judi- cieux des éléments syntaxiques d’UML permette une représentation pertinente des concepts métier et des choix de conception.

Consultant expérimenté et pionnier de l’utilisation d’UML en France, Pascal est aussi un excellent concepteur de cours et un formateur à la pédagogie sans faille. Il n’en est pas non plus à son premier coup d’essai en tant qu’auteur. Après *UML 2 en action*, il nous livre maintenant *UML 2.5 par la pratique*, un excellent recueil de modèles UML issus de projets réels, ou construits pour des études de cas, et minutieusement commentés. Le caractère unique de ce livre réside dans la mise par

296723AWP-UML2\_8E.book Page 8 Lundi, 19. février 2018 7:54 07

**UML 2.5 par la pratique 8**

écrit de commentaires essentiels, qui visent à expliquer un choix de modélisation et qui, généralement, ne sont donnés que par oral sur le « terrain ». Le lecteur pourra ainsi revenir à loisir sur ce qui lui fait souvent défaut, par exemple la justification de l’emploi de tel ou tel trait syntaxique plutôt qu’un autre pour représenter une idée particulière. La mise en parallèle de solutions « alternatives » permet également de comprendre les avantages d’une modélisation sur une autre et la précision séman- tique apportée par la solution choisie.

C’est toute la puissance d’expression d’UML 2 et la complétude de sa syntaxe qui sont démontrées par l’exemple dans cet ouvrage pragmatique et très accessible, que le lecteur, j’en suis certain, prendra plaisir comme moi à lire d’un bout à l’autre. Je me permets cependant de recommander de bien réaliser les exercices complémen- taires, excellents travaux pratiques, mais aussi remarquables tests d’auto-évaluation. En couvrant tous les aspects de l’analyse et de la conception orientée objet, Pascal Roques illustre et explique les variations possibles dans l’usage d’UML pour chacune de ces étapes.

En tant que responsable de la définition de l’offre formation chez Valtech Training, j’ai longtemps eu beaucoup de difficulté à recommander un livre qui serait un bon complément à nos formations à la modélisation objet avec UML. Merci Pascal, grâce à la qualité pédagogique de ton livre, *UML 2.5 par la pratique*, cette tâche est aujourd’hui considérablement plus simple...

Gaël Renault

Ancien Directeur pédagogique de

Valtech Training

296723AWP-UML2\_8E.book Page 9 Lundi, 19. février 2018 7:54 07

Table des matières

**Introduction 13**

Objectifs du livre ................................................................................................................................................. 13 Structure de l’ouvrage ...................................................................................................................................... 14 Conventions typographiques ........................................................................................................................ 16 Remerciements .................................................................................................................................................... 17

PARTIE I – POINT DE VUE FONCTIONNEL

**Chapitre 1 • Modélisation fonctionnelle : étude de cas 21**

Principes et définitions de base .................................................................................................................... 22 Étude d’un guichet automatique de banque .......................................................................................... 25 Étape 1 – Identification des acteurs du GAB .......................................................................................... 26 Étape 2 – Identification des cas d’utilisation .......................................................................................... 29 Étape 3 – Réalisation de diagrammes de cas d’utilisation ................................................................ 30 Étape 4 – Description textuelle des cas d’utilisation ........................................................................... 34 Étape 5 – Description graphique des cas d’utilisation ........................................................................ 40 Étape 6 – Organisation des cas d’utilisation ........................................................................................... 44 Étape 7 – Dynamique globale : *Interaction Overview Diagram* ...................................................... 53

296723AWP-UML2\_8E.book Page 10 Lundi, 19. février 2018 7:54 07

**UML 2.5 par la pratique 10**

**Chapitre 2 • Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques 57**

Étude d’un terminal point de vente ............................................................................................................ 58 Étape 1 – Réalisation du diagramme de cas d’utilisation ................................................................. 59 Étape 2 – Descriptions essentielle et réelle d’un cas d’utilisation ................................................ 63 Étape 3 – Description graphique des cas d’utilisation ........................................................................ 70 Étape 4 – Réalisation d’un diagramme d’états au niveau système ............................................... 75 Conseils méthodologiques ............................................................................................................................. 78

PARTIE II – POINT DE VUE STATIQUE

**Chapitre 3 • Modélisation statique : étude de cas 87**

Principes et définitions de base..................................................................................................................... 88 Étude d’un système de réservation de vol ............................................................................................... 92 Étape 1 – Modélisation des phrases 1 et 2 .............................................................................................. 93 Étape 2 – Modélisation des phrases 6, 7 et 10 ...................................................................................... 95 Étape 3 – Modélisation des phrases 8 et 9 .............................................................................................. 99 Étape 4 – Modélisation des phrases 3, 4 et 5 ......................................................................................... 103 Étape 5 – Ajout d’attributs, de contraintes et de qualificatifs ........................................................ 106 Étape 6 – Utilisation de patterns d’analyse ............................................................................................. 111 Étape 7 – Structuration en packages ......................................................................................................... 114 Étape 8 – Inversion des dépendances ........................................................................................................ 120 Étape 9 – Généralisation et réutilisation .................................................................................................. 122

**Chapitre 4 • Modélisation statique : exercices corrigés et conseils méthodologiques 129**

Compléments sur les relations entre classes .......................................................................................... 130 Modélisation du domaine en pratique ...................................................................................................... 140 Les classes structurées UML 2 ....................................................................................................................... 147 Découverte d’un « pattern » ........................................................................................................................... 156 Conseils méthodologiques ............................................................................................................................. 160

296723AWP-UML2\_8E.book Page 11 Lundi, 19. février 2018 7:54 07

**Sommaire 11**

PARTIE III – POINT DE VUE DYNAMIQUE

**Chapitre 5 • Modélisation dynamique : étude de cas 167**

Principes et définitions de base .................................................................................................................... 168 Étude d’un Publiphone à pièces ................................................................................................................... 173 Étape 1 – Identification des acteurs et des cas d’utilisation ............................................................ 174 Étape 2 – Réalisation du diagramme de séquence système ............................................................. 175 Étape 3 – Représentation du contexte dynamique .............................................................................. 178 Étape 4 – Description exhaustive par un diagramme d’états .......................................................... 181

**Chapitre 6 • Modélisation dynamique : exercices corrigés et conseils méthodologiques 195**

Concepts avancés du diagramme d’états ................................................................................................. 201 Concepts de base du diagramme d’activité ............................................................................................. 217 Concepts avancés du diagramme d’activité ............................................................................................ 223 Conseils méthodologiques .............................................................................................................................. 226

PARTIE IV – CONCEPTION

**Chapitre 7 • Étude de cas complète : de la modélisation métier à la conception détaillée en Java ou C# 233**

Étape 1 – Modélisation métier (business modeling) ............................................................................ 235 Étape 2 – Définition des besoins du système informatique ............................................................. 239 Étape 3 – Analyse du domaine (partie statique) ................................................................................... 248 Étape 4 – Analyse du domaine (partie dynamique) ............................................................................. 262 Étape 5 – Définition des itérations ............................................................................................................. 266 Étape 6 – Définition de l’architecture système ...................................................................................... 267 Étape 7 – Définition des opérations système (itération #1) ............................................................. 272 Étape 8 – Diagrammes d’interaction (itération #1) ............................................................................. 275 Étape 9 – Diagrammes de classes de conception (itération #1) ..................................................... 284 Étape 10 – Définition des opérations système (itérations #2 et #3) ............................................ 291

296723AWP-UML2\_8E.book Page 12 Lundi, 19. février 2018 7:54 07

**UML 2.5 par la pratique 12**

Étape 11 – Contrats d’opérations (itérations #2 et #3) ..................................................................... 295 Étape 12 – Diagrammes d’interaction (itérations #2 et #3) ............................................................ 297 Étape 13 – Diagrammes de classes de conception (itérations #2 et #3) .................................... 298 Étape 14 – Retour sur l’architecture .......................................................................................................... 300 Étape 15 – Passage au code objet ............................................................................................................... 301 Étape 16 – Déploiement de l’application .................................................................................................. 311

**Chapitre 8 • Études de cas complémentaires 313**

Étude du système d’information d’une bibliothèque .......................................................................... 314 Analyse et conception du jeu de démineur ............................................................................................. 329 Conseils méthodologiques ............................................................................................................................. 344 Conclusion ............................................................................................................................................................. 349

**Conclusion 351**

UML, 12 ans après... (7e édition) ................................................................................................................. 352 La modélisation agile avec UML (8e édition) ........................................................................................... 365

ANNEXES

**Annexe 1 • Correspondances UML – Java – C# 375**

La structure statique ......................................................................................................................................... 376 Les relations .......................................................................................................................................................... 380

**Annexe 2 • Glossaire 389**

**Annexe 3 • Bibliographie 399**

Bibliographie des chapitres 1-2 ................................................................................................................... 399 Bibliographie des chapitres 3-4 ................................................................................................................... 400 Bibliographie des chapitres 5-6 ................................................................................................................... 401 Bibliographie des chapitres 7-8 ................................................................................................................... 401

**Index 403**

296723AWP-UML2\_8E.book Page 13 Lundi, 19. février 2018 7:54 07

Introduction

**Objectifs du livre**

Depuis de nombreuses années maintenant, les ouvrages sur le langage UML et la modélisation objet se sont multipliés. *De Merise à UML*1, *UML et SQL*2*, Bases de données avec UML*3, etc., représentent autant de sujets intéressants pour les profes- sionnels de l’informatique. Cependant, ma pratique de la formation (plus d’un millier de personnes formées à OMT, puis UML, depuis 1993...) m’a convaincu qu’il existait encore un besoin qui n’est pas couvert par la multitude d’ouvrages disponibles : un livre d’exercices corrigés4. Je consacre en effet de plus en plus de temps pendant les sessions que j’anime à des séances de discussion avec les stagiaires sur les mérites comparés de telle ou telle solution de modélisation. Et je suis intimement persuadé que ces périodes d’interactivité sur des sujets concrets ont un impact bien plus durable pour eux que la présentation théorique des subtilités du formalisme UML ! Cela m’a amené à constituer une importante base de données d’exercices, rassemblés au fil des années à partir des nombreuses formations que j’ai animées. Je me suis également inspiré des livres fondamentaux qui m’ont aidé personnellement dans mon approfondis- sement de ce sujet, en particulier celui de J. Rumbaugh sur OMT5 (l’un des premiers à proposer des exercices après chaque chapitre de présentation) et le best-seller de C. Larman6 sur l’analyse et la conception objet. C’est ce matériel pédagogique, fondé sur des heures de discussions enrichissantes avec des stagiaires de tous horizons, que je voudrais partager aujourd’hui avec vous. Par leurs

1. *De Merise à UML*, N. Kettani *et al.*, Eyrolles, 1998. 2. *UML et SQL* : *Conception de bases de données,* C. Soutou, Eyrolles, 2002. 3. *Bases de données avec UML*, E.Naiburg R.Maksimchuk, Campus Press, 2002. 4. Mon idée semble avoir fait des émules depuis la parution de la première édition en 2001, puisque d’autres livres du même type sont parus au fil des années : *Exercices corrigés d'UML*, Ellipses 2005, *UML 2 – Initiation, exemples et exer- cices corrigés*, Eni 2008, etc. 5. *Object-Oriented Modeling and Design*, J. Rumbaugh *et al.*, Prentice Hall, 1991. Une version mise à jour est parue der- nièrement en français, sous le titre : *Modélisation et conception orientées objet avec UML2*, Pearson Education, 2005 (mais la magie ne semble plus opérer comme la première fois...). 6. *Applying UML and Patterns*, C. Larman, Prentice Hall, 1997. La troisième édition de cet ouvrage a été traduite en fran-

çais : *UML 2 et les Design Patterns*, Campus Press, 2005.

296723AWP-UML2\_8E.book Page 14 Lundi, 19. février 2018 7:54 07

**UML 14**

**2.5 par la pratique** questions et leurs propositions, ils m’ont forcé à prendre en compte les points de vue les plus divers sur un même problème de modélisation, à améliorer mon argumentation et parfois à envisager de nouvelles solutions auxquelles je n’avais absolument pas pensé ! Il est à noter que cette huitième édition incorpore de nombreux avis d’experts auxquels j’ai demandé leur sentiment sur l’actualité et la pertinence d’UML en 2009, près de quinze ans après sa création. Vous retrouverez ces avis, parfois différents, toujours instructifs, dans des encadrés au fil des chapitres ainsi qu’à la fin de l’ouvrage.

J’ai également voulu évoquer le nouveau langage SysML de modélisation de systèmes, dérivé d’UML 2, qui apporte quelques concepts supplémentaires au diagramme de structure composite7 et au diagramme d’activité. Enfin, j’ai ajouté à la fin de l’ouvrage une section sur la modélisation agile avec UML, ou comment marier la modélisation avec le développement agile.

**Structure de l’ouvrage**

Pour ne pas mélanger les problèmes, le livre est découpé suivant les trois points de vue classiques de modélisation : fonctionnel, statique et dynamique, en insistant pour chacun sur le ou les diagrammes UML prépondérants (les diagrammes entre parenthèses sont moins détaillés que les autres).

**Fonctionnel Fonctionnel Diagramme de Cas d’utilisation**

**(Diagramme de Séquence) (Diagramme d'Activité)**

***3 Axes de modélisation***

**Statique Statique Dynamique Dynamique Diagramme de Classes**

**Diagramme d'États Diagramme de Packages**

**Diagramme d'Activité (Diagramme d’Objets)**

**Diagramme de Séquence Diagramme de Structure Composite**

**(Diagramme de Communication) (Diagramme de Déploiement)**

7. Voir en particulier le chapitre 4. Pour les lecteurs désireux d’en savoir plus sur le langage SysML, se référer à mon ouvrage

sur le sujet : *Modélisation de systèmes complexes avec SysML*, P. Roques, éditions Eyrolles 2013.

296723AWP-UML2\_8E.book Page 15 Lundi, 19. février 2018 7:54 07

**Introduction 15**

Les trois premières parties du livre correspondent donc chacune à un point de vue de modélisation. Pour chaque partie, une étude de cas principale, spécifique, sert de premier chapitre. Avant l’étude de cas, on trouve un rappel des principales définitions utilisées dans la partie concernée. Des exercices complémentaires et un récapitulatif des conseils méthodologiques se trouvent dans un deuxième chapitre. La quatrième partie contient plusieurs études de cas. La première, très détaillée, prend bien sûr en compte les trois points de vue précités, mais couvre également en amont la modélisation métier, et en aval la programmation en Java et C# ! La dernière étude de cas aborde le sujet passionnant des *design patterns*. Retrouvez sur le rabat de la couverture une carte de référence regroupant les principaux schémas UML 28 utilisés dans ce livre. Une table des matières condensée est donnée par la carte ci-après.

8. J’ai utilisé comme référence pour ce livre la toute dernière spécification de l’OMG, à savoir le document formal /17-12-

05.pdf : UML 2.5.1 Specification.

296723AWP-UML2\_8E.book Page 16 Lundi, 19. février 2018 7:54 07

**UML 2.5 par la pratique 16**

**Conventions typographiques**

Pour apporter plus de clarté dans la lecture de ce livre, les exercices et les solutions sont mis en évidence par des polices de caractères et des symboles graphiques différents de la façon suivante :

Quelles sont les entités externes qui interagissent directement avec le GAB ?

Considérons linéairement les phrases de l’énoncé.

Pour guider un peu plus le lecteur, les questions ont un niveau de difficulté évalué de un à quatre, et représenté graphiquement :

De façon ponctuelle, pour rompre la monotonie du texte, j’ai également utilisé des enca- drés « À retenir » pour isoler une note sur une question d’un niveau avancé :

**EXERCICE 1-1. Les acteurs**

**Identifiez les acteurs du GAB.**

question difficile nécessitant une argumentation complexe.

**À RETENIR**

Représentations graphiques d’un acteur

La représentation graphique standard de l’acteur en UML est l’icône appelée « stick man », avec le nom de l’acteur sous le dessin. On peut également montrer un acteur sous la forme rectangulaire d’une classe, avec le mot-clé <<actor>>. Une troisième représentation (inter- médiaire entre les deux premières) est également possible, comme indiqué ci-après.

question facile,

question moyenne,

question assez difficile mettant en jeu des concepts UML avancés,

296723AWP-UML2\_8E.book Page 17 Lundi, 19. février 2018 7:54 07

**Introduction 17**

**Remerciements**

Je tiens en premier lieu à remercier tous ceux qui ont participé au fil des années à l’élaboration des supports de cours UML que j’ai eu l’occasion d’animer, comme Pierre Chouvalidzé, Thibault Cuvillier, Michel Ezran, Patrick Le Go, Franck Vallée, Philippe Riaux, Philippe Dubosq, Yann Le Tanou, Christophe Addinquy, Saïd Lahlouh, Birol Berkem, Fabien Brissoneau, Daniel Rosenblatt, Michel André, Thomas Gil, etc., ainsi que les nombreux autres experts qui ont répondu à mon appel de retour d’expérience : Claude Aubry, Nicolas Belloir, Jean-Michel Bruel (un merci supplémentaire pour l’idée de la carte de référence !), Agusti Canals, Olivier Casse, Thierry Cros, Philippe Desfray, Jean-Claude Grosjean et Marc Tizzano. Je voudrais également remercier tous ceux dont les discussions, les remarques et les suggestions m’ont conduit à améliorer mon argumentation. Je pense tout d’abord à mes nombreux stagiaires, ainsi qu’à mes correspondants lors de prestations de conseil sur l’introduction d’UML dans des projets de tous types. Merci aussi à Éric Sulpice des éditions Eyrolles pour son témoignage renouvelé de confiance et, surtout, qui a su me motiver en me proposant de réaliser ce livre d’exer- cices corrigés. Un grand bravo également à l’équipe des éditrices : Muriel et Sandrine qui ont contribué notablement à la réussite de ce projet. Enfin, un grand merci à Sylvie qui m’apporte tous les jours l’équilibre sans lequel je ne pourrais pas continuer à avancer...

Pascal Roques, décembre 2017 Consultant Formateur pascal.roques@prfc.fr http://www.prfc.fr/

296723AWP-UML2\_8E.book Page 18 Lundi, 19. février 2018 7:54 07

296723AWP-UML2\_8E.book Page 19 Lundi, 19. février 2018 7:54 07

**PARTIE I**

**Point de vue fonctionnel**

296723AWP-UML2\_8E.book Page 20 Lundi, 19. février 2018 7:54 07

296723AWP-UML2\_8E.book Page 21 Lundi, 19. février 2018 7:54 07

**1**

Modélisation fonctionnelle : étude de cas

**Mots-clés**

■ Acteur ■ Contexte statique ■ Cas d’utilisation ■ Acteur principal, acteur secondaire ■ Diagramme de cas d’utilisation ■ Scénario, enchaînement ■ Fiche de description d’un cas d’utilisation ■ Dia- gramme de séquence système ■ Diagramme d’activité ■ Inclusion, extension et généralisation de cas d’utilisation ■ Généralisation/spécialisation d’acteurs ■ Package de cas d’utilisation ■ *Interaction Overview Diagram*.

Ce chapitre va nous permettre d’illustrer pas à pas, sur une première étude de cas, les principales difficultés liées à la mise en œuvre de la technique des cas d’utilisation. Après avoir identifié les acteurs qui interagissent avec le système, nous y développons un premier modèle UML de haut niveau, pour pouvoir établir précisément les frontières du système. Dans cette optique, nous apprenons à identifier les cas d’utilisation et à construire un diagramme reliant les acteurs et les cas d’utilisation. Ensuite, nous préci- sons le point de vue fonctionnel en détaillant les différentes façons dont les acteurs peuvent utiliser le système. À cet effet, nous apprenons à rédiger des descriptions textuelles de cas d’utilisation, ainsi qu’à dessiner des diagrammes UML complémen- taires (comme les diagrammes de séquence ou d’activité).

296723AWP-UML2\_8E.book Page 22 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **22**

**Principes et définitions de base**

**Acteur** Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif

matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l’état du système, en émettant et/ou en recevant des messages susceptibles d’être porteurs de données.

**Comment les identifier ?**

Les acteurs candidats sont systématiquement :

• les utilisateurs humains directs : faites donc en sorte d’identifier 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.

296723AWP-UML2\_8E.book Page 23 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **23**

**Comment les représenter ?**

La représentation graphique standard de l’acteur en UML est l’icône appelée *stick man*, avec le nom de l’acteur sous le dessin. On peut également figurer un acteur sous la forme rectangulaire d’une classe, avec le mot-clé <<actor>>. Une troisième représentation (intermédiaire entre les deux premières) est également possible avec certains outils, comme cela est indiqué ci-après.

Figure 1-1. *Représentations graphiques possibles d’un acteur*

Une bonne recommandation consiste à faire prévaloir l’utilisation de la forme graphique du *stick man* pour les acteurs humains et une représentation rectangulaire pour les systèmes connectés.

**Cas d’utilisation**

Un *cas d*’*utilisation* (« use case ») représente un ensemble de séquences d’actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Chaque cas d’utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire *ce que* le futur système devra faire, sans spécifier *comment* il le fera.

**Comment les identifier ?**

L’objectif est le suivant : l’ensemble des cas d’utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d’utilisation correspond donc à une fonction métier du système, selon le point de vue d’un de ses acteurs.

Pour chaque acteur, il convient de :

• rechercher les différentes intentions métier avec lesquelles il utilise le système,

• déterminer dans le cahier des charges les services fonctionnels attendus du système.

Nommez les cas d’utilisation par un verbe à l’infinitif suivi d’un complément, du point de vue de l’acteur (et non pas du point de vue du système).

296723AWP-UML2\_8E.book Page 24 Lundi, 19. février 2018 7:54 07

**Point 24**

**de vue fonctionnel** *PREMIÈRE PARTIE* **Comment les analyser ?**

Pour détailler la dynamique du cas d’utilisation, la procédure la plus évidente consiste à recenser de façon textuelle 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 cas nominal, les différents cas alternatifs et d’erreur, tout en essayant d’ordonner séquentiellement les descriptions, afin d’améliorer leur lisibilité.

**Comment les représenter ?**

Le diagramme de cas d’utilisation est un schéma qui montre les cas d’utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (icône du « stick man », ou représenta- tion graphique équivalente). Chaque association signifie simplement « participe à ». Un cas d’utilisation doit être relié à au moins un acteur.

Figure 1-2. *Diagramme de cas d’utilisation*

Une fois les cas d’utilisation identifiés, il faut encore les décrire !

**Scénario**

Un *scénario* représente une succession particulière d’enchaînements, s’exécutant du début à la fin du cas d’utilisation, un *enchaînement* étant l’unité de description de séquences d’actions*.* Un cas d’utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent de façon normale) ou d’erreur (qui se terminent en échec).

On peut d’ailleurs proposer une définition différente pour un cas d’utilisation : « ensemble de scénarios d’utilisation d’un système reliés par un but commun du point de vue d’un acteur ».

296723AWP-UML2\_8E.book Page 25 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **25**

Figure 1-3. *Représentation des scénarios d’un cas d’utilisation*

La fiche de description textuelle d’un cas d’utilisation n’est pas normalisée par UML. Nous préconisons pour notre part la structuration suivante :

**Sommaire d’identification (obligatoire)**

**Étude d’un guichet automatique de banque**

Cette étude de cas concerne un système simplifié de Guichet Automatique de Banque (GAB). Le GAB offre les services suivants : 1. Distribution d’argent à tout Porteur de carte de crédit, *via* un lecteur de carte et un

distributeur de billets. 2. Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les

clients porteurs d’une carte de crédit de la banque adossée au GAB.

N’oubliez pas non plus que : 3. Toutes les transactions sont sécurisées. 4. Il est parfois nécessaire de recharger le distributeur, etc.

À partir de ces quatre phrases, nous allons progressivement :

• identifier les acteurs ;

• identifier les cas d’utilisation ;

Inclut titre, résumé, dates de création et de modification, version, responsable, acteurs...

**Description des scénarios (obligatoire)**

Décrit le scénario nominal, les scénarios (ou enchaînements) alternatifs, les scénarios (ou enchaînements) d’erreur, mais aussi les préconditions et les postconditions.

**Exigences non- fonctionnelles (optionnel)**

Ajoute, si c’est pertinent, les informations suivantes : fréquence, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, performances, concurrence, etc. Précise également les contraintes d’interface homme-machine comme des règles d’ergonomie, une charte graphique, etc.

296723AWP-UML2\_8E.book Page 26 Lundi, 19. février 2018 7:54 07

**Point 26**

**de vue fonctionnel** *PREMIÈRE PARTIE* • construire un diagramme de cas d’utilisation ;

• décrire textuellement les cas d’utilisation ;

• compléter les descriptions par des diagrammes dynamiques ;

• organiser et structurer les cas d’utilisation.

L’énoncé précédent est volontairement incomplet et imprécis, comme il en est dans les projets réels ! Notez également que le problème et sa solution sont basés sur l’utilisation de cartes à puce dans le contexte des systèmes bancaires français. Si le GAB que vous avez l’habitude d’utiliser ne fonctionne pas exactement comme le nôtre, ce n’est pas très important. C’est surtout un prétexte pour vous montrer comment raisonner fonctionnellement avec les cas d’utilisation UML.

**Étape 1 – Identification des acteurs du GAB**

Quelles sont les entités externes qui interagissent directement avec le GAB ?

Considérons linéairement les phrases de l’énoncé.

La phrase 1 nous permet d’identifier immédiatement un premier acteur évident : tout « Porteur de carte ». Il pourra uniquement utiliser le GAB pour retirer de l’argent avec sa carte.

En revanche, attention : le lecteur de carte et le distributeur de billets font partie du GAB. Ils ne peuvent donc pas être considérés comme des acteurs ! Vous pouvez noter ici que l’identification des acteurs oblige à fixer précisément la frontière entre le sys- tème à l’étude et son environnement. Si nous restreignions l’étude au système de contrôle-commande des éléments physiques du GAB, le lecteur de carte et le distr- ibuteur de billets deviendraient alors des acteurs.

Autre piège : la carte bancaire elle-même est-elle un acteur ? La carte est bien externe au GAB, et elle interagit avec lui... Pourtant, nous ne recommandons pas de la répertorier en tant qu’acteur, car nous appliquons le principe suivant : éliminer autant que possible les acteurs « physiques » au profit des acteurs « logiques ». L’acteur est celui qui bénéficie de l’utilisation du système. C’est bien le Porteur de carte qui retire de l’argent pour le dépenser ensuite, pas la carte !

**ATTENTION**

**EXERCICE 1-1. Les acteurs**

**Identifiez les acteurs du GAB.**

296723AWP-UML2\_8E.book Page 27 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **27**

La phrase 2 identifie des services supplémentaires qui ne sont proposés qu’aux clients de la banque porteurs d’une carte de crédit de cette dernière. Il s’agit donc d’un profil différent du précédent, que nous matérialisons par un deuxième acteur, appelé *Client banque*1.

La phrase 3 nous incite à prendre en compte le fait que toutes les transactions sont sécurisées. Mais sécurisées par qui ? Pas par le GAB. Il existe donc d’autres entités externes qui jouent le rôle de Système d’autorisation et avec lesquelles le GAB com- munique directement. Une interview de l’expert métier est nécessaire, pour nous permettre d’identifier deux acteurs différents :

• le Système d’autorisation global Carte Bancaire, pour les transactions de retrait ;

• le Système d’information de la banque, pour autoriser toutes les transactions effectuées par un client avec sa carte de la banque, mais également pour accéder au solde des comptes.

Enfin, la phrase 4 nous rappelle qu’un GAB nécessite également des actions de mainte- nance, telles que le rechargement en billets du distributeur, la récupération des cartes avalées, etc. Ces actions de maintenance sont effectuées par un nouvel acteur, que nous appellerons pour simplifier : *Opérateur de maintenance*2.

Plutôt que de répertorier simplement les acteurs textuellement, on peut réaliser un pre- mier diagramme que nous appelons *diagramme de contexte statique*. Il suffit pour cela d’utiliser un diagramme de classes dans lequel chaque acteur est relié par une associa- tion à une classe centrale unique représentant le système, ce qui permet en outre de spé- cifier le nombre d’instances d’acteurs connectées au système à un moment donné.

Bien que ce diagramme ne fasse pas partie des diagrammes UML « officiels », nous l’avons très souvent trouvé utile dans notre expérience des projets réels.

Le GAB est un système fondamentalement mono-utilisateur : à tout instant, il n’y a qu’une instance de chaque acteur (au maximum) connectée au système.

1. Il faudrait parler en toute rigueur de « Porteur de carte client de la banque », mais c’est un peu long, d’où la nécessité

absolue de documenter tout élément UML, y compris les acteurs. 2. Nous pourrions par exemple identifier un acteur supplémentaire appelé « *Convoyeur* » ou « *Gabiste* », chargé spécifi-

quement de remplir la caisse du GAB.

**EXERCICE 1-2. Diagramme de contexte statique**

**Élaborez le diagramme de contexte statique du GAB.**

296723AWP-UML2\_8E.book Page 28 Lundi, 19. février 2018 7:54 07

**Point 28**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 1-4. *Diagramme de contexte statique du GAB*

Il faudrait en toute rigueur ajouter une note graphique pour indiquer qu’en outre les acteurs humains *Client banque* et *Porteur de carte* sont mutuellement exclusifs, ce qui n’est pas implicite d’après les multiplicités des associations.

Figure 1-5. *Diagramme de contexte statique du GAB complété*

Une autre solution, un peu plus élaborée, consiste à considérer que *Client banque* est une spécialisation de *Porteur de carte*, comme cela est illustré sur la figure sui- vante. Le problème précité d’exclusivité est ainsi résolu par construction, grâce à l’héritage entre acteurs.

296723AWP-UML2\_8E.book Page 29 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **29**

Figure 1-6. *Version plus élaborée du diagramme de contexte statique du GAB*

**Étape 2 – Identification des cas d’utilisation**

**EXERCICE 1-3. Cas d’utilisation**

**Préparez une liste préliminaire des cas d’utilisation du GAB, par acteur.**

Reprenons un à un les cinq acteurs et listons les différentes façons qu’ils ont d’utiliser le GAB :

Porteur de carte :

• Retirer de l’argent.

Client banque :

• Retirer de l’argent (à ne pas oublier !).

• Consulter le solde de son compte courant.

• Déposer du numéraire.

• Déposer de l’argent (du numéraire ou des chèques)3.

Opérateur de maintenance :

• Recharger le distributeur.

• Maintenir l’état opérationnel (récupérer les cartes avalées, récupérer les chèques déposés, remplacer le ruban de papier, etc.).

3. Il n’est pas nécessaire de répertorier deux cas d’utilisation distincts appelés *Déposer du numéraire* et *Déposer des chèques*. En effet, ils auraient le même acteur principal et le même objectif global. Cependant, cette décision pourra être remise en cause lors de la description détaillée des cas d’utilisation si l’on s’aperçoit alors que les scénarios sont trop différents. N’oubliez pas qu’une décision de modélisation ne doit jamais être irréversible !

296723AWP-UML2\_8E.book Page 30 Lundi, 19. février 2018 7:54 07

**Point 30**

**de vue fonctionnel** *PREMIÈRE PARTIE* Système d’autorisation (Sys. Auto.) :

• Néant.

Système d’information (SI) banque :

• Néant.

**À RETENIR**

Acteur principal ou secondaire

Contrairement à ce que l’on pourrait croire, tous les acteurs n’utilisent pas forcément le sys- tème ! Nous appelons acteur principal celui pour qui le cas d’utilisation produit un résultat observable. Par opposition, nous qualifions d’acteurs secondaires les autres participants du cas d’utilisation. Les acteurs secondaires sont souvent sollicités pour des informations complémen- taires ; ils peuvent uniquement consulter ou informer le système lors de l’exécution du cas d’uti- lisation.

C’est exactement le cas des deux acteurs « non humains » de notre exemple : le Sys. Auto. et le SI banque sont uniquement sollicités par le GAB dans le cadre de la réalisation de certains cas d’utilisation. Mais ils n’ont pas eux-mêmes de façon propre d’utiliser le GAB, d’objectif à part entière.

**Étape 3 – Réalisation de diagrammes de cas d’utilisation**

On obtient sans difficulté un diagramme préliminaire en transcrivant la réponse précé- dente sur un schéma qui montre les cas d’utilisation (ovales) reliés par des associations (lignes) à leurs acteurs principaux (icône du « stick man »).

**À RETENIR**

Cadre de diagramme : tag et nom

Notez que depuis UML 2.0, un diagramme peut être inclus dans un cadre accueillant tout le contenu graphique. Le cadre a pour intitulé le nom du diagramme et établit sa portée. C’est un rectangle avec un petit pentagone (appelé tag de nom) placé dans l’angle supérieur gauche, qui contient le type du diagramme et son nom. Le cadre n’est cependant pas obliga- toire lorsque le contexte est clair. La spécification UML définit les tags de chaque type de diagramme, mais cela n’a pas de caractère obligatoire et chaque outil a fait ses propres choix. Celui que nous avons utilisé majoritairement pour les nouveaux diagrammes UML 2 du livre, Enterprise Architect de la société Sparx Systemsa, a pris le parti de définir tous les tags avec deux lettres : ud, cd, sd, td, ad, sm, id, dd. L’autre outil que nous avons égale- ment utilisé, MagicDraw, prend des conventions parfois légèrement différentes. Dans le dia- gramme suivant, uc signifie *use ca*se *diagram.*

a. Le lecteur peut visiter le site web suivant http://www.sparxsystems.com.au/ où il trouvera une version d’évalua- tion du produit *Enterprise Architect*. Nous recommandons également l’introduction à UML 2 disponible sur le même site.

296723AWP-UML2\_8E.book Page 31 Lundi, 19. février 2018 7:54 07

Figure 1-8. *Version plus sophistiquée du diagramme de cas d’utilisation préliminaire*

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **31**

Figure 1-7. *Diagramme de cas d’utilisation préliminaire du GAB*

**EXERCICE 1-4. Généralisation entre acteurs**

**Proposez une** *autre* **version, plus sophistiquée, de ce diagramme de cas d’utilisation préliminaire.**

Le cas d’utilisation *Retirer de l*’*argent* a deux acteurs principaux possibles (mais exclusifs du point de vue de la simultanéité). Une autre façon de l’exprimer consiste à considérer l’acteur *Client de la banque* comme une spécialisation (au sens de la relation d’héritage) de l’acteur plus général *Porteur de carte*, comme nous l’avons déjà indiqué sur la figure 1-6. Un client de la banque est en effet un Porteur de carte particulier qui a toutes les prérogatives de ce dernier, ainsi que d’autres qui lui sont propres en tant que client.

UML permet de décrire une relation de généralisation/spécialisation entre acteurs, comme cela est indiqué sur le diagramme de cas d’utilisation suivant.

296723AWP-UML2\_8E.book Page 32 Lundi, 19. février 2018 7:54 07

**Point 32**

**de vue fonctionnel** *PREMIÈRE PARTIE* Cependant, dans notre exemple, l’intérêt de cette relation de généralisation n’est pas évident. Elle permet certes de supprimer l’association entre l’acteur *Client banque* et le cas d’utilisation *Retirer de l*’*argent*, qui est maintenant héritée de l’acteur *Porteur de carte*, mais ajoute en revanche le symbole de généralisation entre les deux acteurs... De plus, nous verrons au paragraphe suivant que les acteurs secondaires sollicités ne sont pas les mêmes dans le cas du Porteur de carte non client et dans celui du client de la banque.

Nous ne retiendrons donc pas cette solution et nous considérerons dans la suite du chapitre que les dénominations *Porteur de carte non client* et *Porteur de carte* sont synonymes.

Il nous reste maintenant à ajouter les acteurs secondaires pour compléter le dia- gramme de cas d’utilisation. Pour cela, UML propose simplement en standard de faire apparaître ces acteurs avec des associations supplémentaires vers les cas d’uti- lisation existants.

Précisions graphiques au diagramme de cas d’utilisation

Pour notre part, afin d’améliorer le contenu informatif de ces diagrammes, 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.

Pour tous les cas d’utilisation propres au client de la banque, il faut clairement faire intervenir comme acteur secondaire *SI banque*.

Mais un problème se pose pour le cas d’utilisation partagé *Retirer de l*’*argent*. En effet, si l’acteur principal est un Porteur de carte non client, il faudra faire appel au *Sys. Auto.* (qui se chargera ensuite de contacter le SI de la banque du porteur), alors que, s’il s’agit d’un client de la banque, le GAB contactera directement le *SI banque*.

Une première solution consiste à ajouter une association avec chacun des deux acteurs non-humains. Cette modélisation simpliste ne permet pas au lecteur du diagramme de

**À RETENIR**

**EXERCICE 1-5. Acteurs secondaires**

**Complétez le diagramme de cas d’utilisation préliminaire en ajoutant les acteurs secondaires. Pour simplifier, ne tenez plus compte pour l’instant de l’opérateur de maintenance.**

296723AWP-UML2\_8E.book Page 33 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **33**

comprendre que les acteurs participent au cas d’utilisation *Retirer de l*’*argent* sélective- ment deux par deux et non pas tous ensemble (figure 1-9).

Figure 1-9. *Version simple du diagramme de cas d’utilisation complété*

Une autre solution consiste à distinguer deux cas d’utilisation pour le retrait d’argent : *Retirer de l*’*argent* et *Retirer de l*’*argent avec une carte de la banque*. Cette modélisation plus précise, mais plus lourde, est plus parlante pour l’expert métier. Elle milite d’ail- leurs clairement contre l’utilisation de la généralisation entre acteurs évoquée précédem- ment. En effet, la distinction entre les deux cas d’utilisation est contradictoire avec la tentative d’héritage par l’acteur *Client banque* du cas unique *Retirer de l*’*argent*, qui avait été envisagée plus haut, alors que les acteurs secondaires n’avaient pas encore été ajoutés. Nous garderons cette seconde solution pour la suite de l’exercice4 (figure 1-10).

Figure 1-10. *Fragment de la version plus précise du diagramme de cas d’utilisation complété*

On notera que le *SI banque* n’est pas un acteur direct du cas d’utilisation *Retirer de l*’*argent*, car nous considérons que le *Sys. Auto.* se charge de le contacter, en dehors de la portée du GAB.

4. Il s’agit ici d’un choix de modélisation arbitraire ! Nous ne disons pas que toute autre solution serait mauvaise, mais

nous expliquons avec des arguments concrets pourquoi nous préférons la nôtre.

296723AWP-UML2\_8E.book Page 34 Lundi, 19. février 2018 7:54 07

**Point 34**

**de vue fonctionnel** *PREMIÈRE PARTIE* **Étape 4 – Description textuelle des cas d’utilisation**

**EXERCICE 1-6. Partie obligatoire du cas d’utilisation**

**Décrivez la partie obligatoire du cas d’utilisation RETIRER DE L’ARGENT (pour l’acteur non client de la banque).**

**Sommaire d’identification**

**Titre** : Retirer de l’argent

**Résumé** : ce cas d’utilisation permet à un Porteur de carte, qui n’est pas client de la banque, de retirer de l’argent, si son crédit hebdomadaire le permet.

**Acteurs** : Porteur de carte (principal), *Système d’autorisation (secondaire).*

**Date de création** : 02/03/17 **Date de mise à jour** : 05/05/17

**Version** : 1.8 **Responsable** : Pascal Roques

**Description des scénarios**

**Préconditions**

• La caisse du GAB est alimentée (il reste au moins un billet !).

• Aucune carte ne se trouve déjà coincée dans le lecteur.

• La connexion avec le Système d’autorisation est opérationnelle.

**Scénario nominal** 1. Le Porteur de carte5 introduit sa carte dans le lecteur de cartes du GAB. 2. Le GAB vérifie que la carte introduite est bien une carte bancaire. 3. Le GAB demande au Porteur de carte de saisir son code d’identification. 4. Le Porteur de carte saisit son code d’identification. 5. Le GAB compare le code d’identification avec celui qui est codé sur la puce de

la carte. 6. Le GAB demande une autorisation au Système d’autorisation. 7. Le Système d’autorisation donne son accord et indique le crédit hebdomadaire. 8. Le GAB demande au Porteur de carte de saisir le montant désiré du retrait. 9. Le Porteur de carte saisit le montant désiré du retrait.

5. Nous préconisons de mettre systématiquement une majuscule au début du nom des acteurs pour améliorer la lisibilité du

scénario nominal.

296723AWP-UML2\_8E.book Page 35 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **35**

10. Le GAB contrôle le montant demandé par rapport au crédit hebdomadaire. 11. Le GAB demande au Porteur de carte s’il veut un ticket. 12. Le Porteur de carte demande un ticket. 13. Le GAB rend sa carte au Porteur de carte. 14. Le Porteur de carte reprend sa carte. 15. Le GAB délivre les billets et un ticket. 16. Le Porteur de carte prend les billets et le ticket. Une autre présentation intéressante6 consiste à séparer les actions des acteurs et du système en deux colonnes comme suit :

1. Le Porteur de carte introduit sa carte

2. Le GAB vérifie que la carte introduite dans le lecteur de cartes du GAB.

est bien une carte bancaire. 3. Le GAB demande au Porteur de carte de

saisir son code d’identification.

4. Le Porteur de carte saisit son code

5. Le GAB compare le code d’identification d’identification.

avec celui qui est codé sur la puce de la carte. 6. Le GAB demande une autorisation au

Système d’autorisation.

7. Le Système d’autorisation donne son accord et indique le crédit hebdoma- daire.

6. Cette présentation a été recommandée par C. Larman dans la première version de *Applying UML and Patterns [Larman 97]*.

8. Le GAB demande au Porteur de carte de

saisir le montant désiré du retrait.

9. Le Porteur de carte saisit le montant

désiré du retrait.

10. Le GAB contrôle le montant demandé par

rapport au crédit hebdomadaire. 11.Le GAB demande au Porteur de carte

s’il veut un ticket.

12. Le Porteur de carte demande un ticket. 13. Le GAB rend sa carte au Porteur de carte.

14. Le Porteur de carte reprend sa carte. 15. Le GAB délivre les billets et un ticket.

16. Le Porteur de carte prend les billets et le

ticket.

296723AWP-UML2\_8E.book Page 36 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **36**

**Enchaînements alternatifs7**

*A1 : code d*’*identification provisoirement erroné*

L’enchaînement A1 démarre au point 5 du scénario nominal. 6. Le GAB indique au Porteur de carte que le code est erroné, pour la première ou

deuxième fois. 7. Le GAB enregistre l’échec sur la carte.

Le scénario nominal reprend au point 3.

*A2 : montant demandé supérieur au crédit hebdomadaire*

L’enchaînement A2 démarre au point 10 du scénario nominal. 11. Le GAB indique au Porteur de carte que le montant demandé est supérieur au

crédit hebdomadaire.

Le scénario nominal reprend au point 8.

*A3 : ticket refusé*

L’enchaînement A3 démarre au point 11 du scénario nominal. 12. Le Porteur de carte refuse le ticket. 13. Le GAB rend sa carte au Porteur de carte. 14. Le Porteur de carte reprend sa carte. 15. Le GAB délivre les billets. 16. Le Porteur de carte prend les billets.

**Enchaînements d’erreur**

*E1 : carte non-valide*

L’enchaînement E1 démarre au point 2 du scénario nominal. 3. Le GAB indique au Porteur que la carte n’est pas valide (illisible, périmée, etc.),

la confisque ; le cas d’utilisation se termine en échec.

*E2 : code d*’*identification définitivement erroné*

L’enchaînement E2 démarre au point 5 du scénario nominal. 6. Le GAB indique au Porteur de carte que le code est erroné, pour la troisième

fois. 7. Le GAB confisque la carte. 8. Le Système d’autorisation est informé ; le cas d’utilisation se termine en échec.

7. Nous distinguons les enchaînements alternatifs (Ax) qui reprennent ensuite à une étape du scénario nominal des enchaî- nements d’erreur (Ey) qui terminent brutalement le cas d’utilisation en échec. L’objectif de l’acteur principal est donc atteint par les scénarios nominaux et alternatifs mais pas par ceux d’erreur.

296723AWP-UML2\_8E.book Page 37 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **37**

*E3 : retrait non autorisé* L’enchaînement E3 démarre au point 6 du scénario nominal. 7. Le Système d’autorisation interdit tout retrait. 8. Le GAB éjecte la carte ; le cas d’utilisation se termine en échec. *E4 : carte non reprise* L’enchaînement E4 démarre au point 13 du scénario nominal. 14. Au bout de 10 secondes, le GAB confisque la carte. 15. Le Système d’autorisation est informé ; le cas d’utilisation se termine en échec. *E5 : billets non pris* L’enchaînement E5 démarre au point 15 du scénario nominal. 16. Au bout de 10 secondes, le GAB reprend les billets. 17. Le cas d’utilisation se termine en échec. *E6 : annulation de la transaction* L’enchaînement E6 peut démarrer entre les points 4 et 12 du scénario nominal. 4 à 12. Le Porteur de carte demande l’annulation de la transaction en cours.

Le GAB éjecte la carte ; le cas d’utilisation se termine en échec.

Une autre présentation intéressante des enchaînements alternatifs et d’erreur consiste à utiliser les conventions préconisées par A. Cockburn8. Celui-ci pro- pose d’indiquer les différentes alternatives par des lettres collées au chiffre du numéro de l’étape du scénario nominal concernée. Une version alternative de la solution précédente pourrait être alors : 2a. Carte illisible ou non valable :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 2b. Carte périmée :

Le GAB avertit le Porteur et confisque la carte ; le cas d’utilisation se ter- mine en échec. 4a. Délai de saisie du code expiré :

Le GAB avertit le porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 4-12a.9 Le Porteur annule la transaction :

Le GAB éjecte la carte ; le cas d’utilisation se termine en échec.

8. Le lecteur se référera avec profit à l’excellent ouvrage d’Alistair Cockburn : *Rédiger des cas d’utilisation efficaces*,

éditions Eyrolles 2001 [Cockburn 01]. 9. La notation 4-12 signifie : « de l’étape 4 à l’étape 12 ».

296723AWP-UML2\_8E.book Page 38 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **38**

5a. Code d’identification erroné pour la première ou deuxième fois :

5a1. Le GAB enregistre l’échec sur la carte. 5a2. Le GAB avertit le Porteur et le scénario nominal reprend à l’étape 3. 5b. Code d’identification erroné pour la troisième fois :

Le GAB avertit le Porteur et confisque la carte ; le cas d’utilisation se ter- mine en échec.

7a. Transaction refusée par le Système d’autorisation :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 7b. Délai de réponse du Système d’autorisation expiré :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 9a. Délai de saisie du montant expiré :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 10a. Montant demandé supérieur au crédit hebdomadaire :

Le GAB avertit le Porteur et le scénario nominal reprend à l’étape 8. 10b. Crédit hebdomadaire insuffisant :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec. 12a. Le Porteur ne demande pas de ticket :

Le cas d’utilisation continue à l’identique, sauf l’impression du ticket. 14a. Délai de retrait de la carte expiré :

14a1. Le GAB confisque la carte et annule la transaction ; 14a2. Le GAB avertit le Système d’autorisation et le cas d’utilisation se termine

en échec. 16a. Délai de retrait des billets expiré :

Le GAB confisque les billets et annule la transaction ; le cas d’utilisation se termine en échec. 1-7a. Coupure réseau avec le Système d’autorisation :

Le GAB avertit le Porteur et éjecte la carte ; le cas d’utilisation se termine en échec.

**Postconditions**

La caisse du GAB contient moins de billets qu’au début du cas d’utilisation (le nombre de billets manquants est fonction du montant du retrait).

296723AWP-UML2\_8E.book Page 39 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **39**

Une transaction de retrait a été enregistrée par le GAB avec toutes les informations pertinentes (montant, numéro de carte, date, etc.). Les détails de la transaction doivent être enregistrés aussi bien en cas de succès que d’échec.

**EXERCICE 1-7. Paragraphes optionnels du cas d’utilisation**

**Complétez la description du cas d’utilisation RETIRER DE L’ARGENT avec les paragraphes optionnels. Détaillez les besoins en interface homme-machine.**

**Exigences non fonctionnelles10**

**Contraintes Descriptif**

Temps de réponse

**Besoins d’IHM**

Les dispositifs d’entrée/sortie à la disposition du Porteur de carte doivent être :

• Un lecteur de carte bancaire.

• Un clavier numérique (pour saisir son code), avec des touches « validation », « correction » et « annulation ».

• Un écran pour l’affichage des messages du GAB.

• Des touches autour de l’écran pour sélectionner un montant de retrait parmi ceux qui sont proposés.

• Un distributeur de billets.

• Un distributeur de tickets.

L’interface du GAB doit réagir en l’espace de 2 secondes au maximum. Une transaction nominale de retrait doit durer moins de 2 minutes.

Concurrence Non applicable (mono-utilisateur).

Disponibilité Le GAB est accessible 7 jours sur 7, 24 h sur 24 (global10).

L’absence de papier pour imprimer les tickets ne doit pas empêcher les retraits.

Intégrité Les interfaces du GAB doivent être très robustes pour prévenir le

vandalisme.

Confidentialité La comparaison du code d’identification saisi sur le clavier du GAB

avec celui de la carte doit être fiable à 10– 6.

10. Cette exigence non fonctionnelle n’est pas réellement spécifique au cas d’utilisation et devra donc se retrouver au final

dans un document plus global. Elle a uniquement été citée ici (ainsi que les suivantes) pour étoffer le paragraphe.

296723AWP-UML2\_8E.book Page 40 Lundi, 19. février 2018 7:54 07

**Point 40**

**de vue fonctionnel** *PREMIÈRE PARTIE* **Étape 5 – Description graphique des cas d’utilisation**

Pour documenter les cas d’utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement avec les utilisateurs et de s’entendre sur la terminologie métier employée.

En revanche, le texte présente des désavantages puisqu’il est difficile de montrer comment les enchaînements se succèdent, ou à quel moment les acteurs secondaires sont sollicités. En outre, la maintenance des évolutions s’avère souvent fastidieuse. Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML.

**À RETENIR**

Descriptions dynamiques d’un cas d’utilisation

Pour les cas d’utilisation, on peut utiliser le diagramme d’activité car les utilisateurs le compren- nent d’autant plus facilement qu’il paraît ressembler à un organigramme traditionnelb.

Figure 1-11. *Bases du diagramme d’activité UML 2*

Pour des scénarios particuliers, le diagramme de séquence est une bonne solution.

b. UML 2 propose un nouveau type de diagramme, appelé « Interaction Overview Diagram », qui fusionne les diagrammes d’activité et de séquence. L’utilisation de ce nouveau diagramme paraît prometteuse au niveau des cas d’utilisation, voire du système global. Nous en donnerons un exemple dans la suite du chapitre.

296723AWP-UML2\_8E.book Page 41 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **41**

Figure 1-12. *Bases du diagramme de séquence UML 2*

Nous recommandons de le présenter en montrant l’acteur principal à gauche, puis un objet unique représentant le système en boîte noire, et, enfin, les éventuels acteurs secondaires sollicités durant le scénario à droite du système. Nous utiliserons l’inti- tulé diagramme de séquence système comme cela a été proposé initialement dans [Larman 97]. Avec culier les les intéressants cadres d’interactions ajouts au (avec diagramme les opérateurs de séquence loop, apportés opt et par alt UML par 2, exemple), en parti-

ainsi que la possibilité de référencer une interaction décrite par ailleurs, le diagramme de séquence système nous semble constituer une excellente solution.

**EXERCICE 1-8. Diagramme de séquence système**

**Réalisez un diagramme de séquence système qui décrit le scénario nominal du cas d’utilisation RETIRER DE L’ARGENT.**

Il suffit de transcrire sous forme de diagramme de séquence les interactions citées dans le scénario textuel de la réponse 1-6, en utilisant les conventions graphiques citées plus haut :

• l’acteur principal *Porteur de carte* à gauche ;

• un participant représentant le GAB au milieu ;

• l’acteur secondaire *Sys. Auto.* à droite du GAB.

296723AWP-UML2\_8E.book Page 42 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **42**

Contrairement au diagramme de séquence précédent qui ne décrit que le scénario nominal, le diagramme d’activité doit représenter l’ensemble des actions réalisées par le système, avec tous les branchements conditionnels et toutes les boucles possibles.

C’est un graphe orienté d’actions et de transitions. Les transitions sont franchies lors de la fin des actions ; des étapes peuvent être réalisées en parallèle ou en séquence.

Le diagramme d’activité est décrit sur la figure suivante avec un repérage des principaux symboles graphiques. Notez que le diagramme diffère légèrement du texte : il omet l’étape de demande de tic- ket dans un souci de simplification. Néanmoins, le résultat de cette demande est pris en compte par la condition de garde avec le label « ticket demandé ».

Figure 1-13. *Diagramme de séquence système du scénario nominal de* Retirer de l’argent

**EXERCICE 1-9. Diagramme d’activité**

**Réalisez un diagramme d’activité qui décrit la dynamique du cas d’utilisation RETIRER DE L’ARGENT.**

296723AWP-UML2\_8E.book Page 43 Lundi, 19. février 2018 7:54 07

Figure 1-14. *Diagramme d*’*activité de* Retirer de l’argent**À RETENIR**

**EXERCICE 1-10. Diagramme de séquence système enrichi**

Ajouts au diagramme de séquence système

Une possibilité intéressante consiste à enrichir le diagramme de séquence système du scénario nominal pour 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). Cela donne souvent un diagramme moins complexe à lire que ne l’est un diagramme d’activité, car moins riche en symboles, mais au contenu informatif appréciable pour l’expert métier.

**Enrichissez le diagramme de séquence système qui décrit le scénario nominal du cas d’utilisation RETIRER DE L’ARGENT.**

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **43**

296723AWP-UML2\_8E.book Page 44 Lundi, 19. février 2018 7:54 07

**Point 44**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 1-15. *Diagramme de séquence système enrichi du scénario nominal de* Retirer de l’argent

**Étape 6 – Organisation des cas d’utilisation**

Dans cette avant-dernière étape, nous allons affiner nos diagrammes et nos descriptions.

Avec UML, il est en effet possible de détailler et d’organiser les cas d’utilisation de deux façons différentes et complémentaires :

• en ajoutant des relations d’inclusion, d’extension et de généralisation entre cas d’utilisation ;

• en les regroupant en packages, afin de définir des blocs fonctionnels de plus haut niveau.

Abordons tout d’abord la relation *d*’*inclusion* : le cas d’utilisation de base en incor- pore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements. On utilise cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d’utilisa- tion à part.

296723AWP-UML2\_8E.book Page 45 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **45**

**EXERCICE 1-11. Inclusion entre cas d’utilisation**

**Identifiez une partie commune aux différents cas d’utilisation et factorisez-la dans un nouveau cas inclus dans ces derniers.**

Si l’on examine en détail la description textuelle du cas d’utilisation *Retirer de l*’*argent*, on s’aperçoit que le début du scénario nominal va également être appli- cable à tous les cas d’utilisation du client de la banque, en remplaçant « Porteur de carte » par « Client de la banque » : 1. Le Porteur de carte introduit sa carte dans le lecteur de cartes du GAB. 2. Le GAB vérifie que la carte introduite est bien une carte bancaire. 3. Le GAB demande au Porteur de carte de saisir son code d’identification. 4. Le Porteur de carte saisit son code d’identification. 5. Le GAB compare le code d’identification avec celui qui est codé sur la puce de

la carte.

Cet enchaînement nominal est en outre complété par les enchaînements alternatifs ou d’erreur A1 (code d’identification provisoirement erroné), E1 (carte non valide) et E2 (code d’identification définitivement erroné).

On peut donc légitimement identifier un nouveau cas d’utilisation11 inclus dans les précédents, que nous appellerons *S*’*authentifier*, et qui contient les enchaînements cités plus haut. Cela nous permettra d’enlever des autres cas d’utilisation toutes ces descriptions textuelles redondantes, en se concentrant mieux sur leurs spécificités fonctionnelles.

En UML, cette relation d’inclusion obligatoire est formalisée par une flèche de dépendance entre le cas d’utilisation de base et le cas inclus, nommée avec le mot- clé <<include>>, comme cela est indiqué sur le schéma suivant.

11. Jacobson a proposé dans un article de parler de fragment plutôt que de cas d’utilisation à part entière, car s’authentifier n’est pas un objectif à part entière de l’utilisateur du GAB ! Pour renforcer ce concept, nous préconisons d’utiliser un mot-clé (ou stéréotype) « fragment » pour le différencier des autres cas d’utilisation.

296723AWP-UML2\_8E.book Page 46 Lundi, 19. février 2018 7:54 07

**Point 46**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 1-16. *Relation d’inclusion entre cas d’utilisation*

L’identification du fragment S’authentifier permet d’alléger le diagramme de séquence système en utilisation le cadre ref proposé par UML 2. Le début du diagramme remanié est donné sur la figure 1-17. Notez la contrainte temporelle sur la réponse du système d’autorisation.

Figure 1-17. *Nouveau diagramme de séquence système avec référence au cas inclus*

Le cadre ref fait référence à un autre diagramme de séquence donné figure 1-18.

296723AWP-UML2\_8E.book Page 47 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **47**

Poursuivons notre analyse par la relation *d*’*extension* : cette fois-ci, le cas de base en incorpore implicitement un autre, mais de façon optionnelle, à un endroit spécifié indi- rectement dans celui qui procède à l’extension. On utilise cette relation pour séparer un comportement optionnel ou rare du comportement obligatoire.

.

En réexaminant la question du retrait d’argent, on a tôt fait de s’apercevoir que le client de la banque applique quasiment le même enchaînement nominal que le Porteur de carte. Mais, en tant que client, il a également accès aux autres cas d’utilisation : pourquoi ne pas lui permettre de consulter son solde juste avant qu’il ne choisisse le montant de son retrait ? Il pourrait ainsi ajuster le montant demandé avec ce qu’il lui reste à ce moment sur son compte.

Si l’on retient ce nouveau besoin fonctionnel, pour le modéliser en UML il suffit d’ajouter une relation d’extension optionnelle comme cela est indiqué sur la figure suivante.

Les deux cas d’utilisation peuvent bien sûr s’exécuter indépendamment, mais *Consulter le solde* peut également venir s’intercaler à l’intérieur de *Retirer de l*’*argent avec une carte de la banque*, au point d’extension *Vérification montant*.

Figure 1-18. *Diagramme de séquence système du fragment référencé*

**EXERCICE 1-12. Extension entre cas d’utilisation**

**En extrapolant sur les besoins initiaux, identifiez une relation d’extension entre deux cas d’utilisation du client de la banque.**

296723AWP-UML2\_8E.book Page 48 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **48**

Ce point d’extension doit être déclaré dans la description textuelle, par exemple en modifiant comme ceci l’enchaînement nominal : ...7. Le SI banque donne son accord et indique le solde hebdomadaire. 8. Le GAB demande au Client banque de saisir le montant désiré du retrait.

*Point d*’*extension : Vérification montant* 9. Le Client banque saisit le montant désiré du retrait. 10. Le GAB contrôle le montant demandé par rapport au solde hebdomadaire. ...Poursuivons enfin par la relation de *généralisation / spécialisation* : les cas d’utili- sation descendants héritent de la description de leur parent commun. Ils peuvent néanmoins comprendre chacune des interactions spécifiques supplémentaires, ou modifier les interactions dont ils ont hérité. On utilise cette relation pour formaliser des variations importantes12 sur le même cas d’utilisation.

Considérons le cas d’utilisation *Déposer de l’argent*. Il possède deux scénarios princi- paux : *Déposer du numéraire* et *Déposer des chèques*. Reprenons la discussion enta- mée en début de chapitre : est-il souhaitable de distinguer ces scénarios en tant que cas d’utilisation à part entière ? Essayons de trouver les arguments pour et contre.

**EXERCICE 1-13. Généralisation / spécialisation entre cas d’utilisation**

**Identifiez une relation de généralisation / spécialisation qui implique un cas d’utili- sation du client de la banque.**

12. Nous insistons sur le mot « important », car cette relation de généralisation entre cas d’utilisation se révèle souvent inu- tile dans la pratique, voire dangereuse. Elle ajoute de la complexité au diagramme, ce qui peut rebuter l’expert métier qui doit le valider, surtout si le modélisateur utilise également l’inclusion et l’extension !

Figure 1-19. *Relation d’extension entre cas d’utilisation*

296723AWP-UML2\_8E.book Page 49 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **49**

Ils mettent en jeu les mêmes acteurs : le *Client banque* comme acteur principal et le *SI banque* comme acteur secondaire. Mais surtout, ils parlent de la même chose : la possibilité offerte à un client de la banque d’effectuer un dépôt d’argent grâce au GAB. Le fait que cette transaction consiste à glisser des billets dans un lecteur de billets, ou à simplement déposer une enveloppe contenant un ou plusieurs chèques, n’est pas fondamental. Le résultat sera similaire, à savoir une ligne de crédit sur le compte du client.

Pourtant, le détail des enchaînements va varier notablement : le dépôt de numéraire implique par exemple un dispositif de reconnaissance de billets, avec des interac- tions liées à chaque introduction de billets, aux erreurs possibles (billet non reconnu, etc.) et à la fin de la transaction. Il est aussi probable que le système de tenue des comptes (qui fait partie du *SI banque*) soit informé en temps réel du dépôt afin de créditer le compte, alors que le dépôt de chèques donnera lieu pour sa part à une vérification manuelle par un guichetier bien après que la transaction fut terminée. En distinguant deux cas d’utilisation, nous ajoutons la possibilité de leur associer des acteurs secondaires différents.

Pour formaliser cette unité fonctionnelle, tout en se gardant la possibilité de décrire les différences au niveau des enchaînements, nous pouvons utiliser la relation de généralisation / spécialisation. Il suffit de considérer que *Déposer de l’argent* est un cas d’utilisation généralisé. Ce cas a maintenant la particularité d’être abstrait (il apparaît alors en italiques), car il ne s’instancie pas directement, mais uniquement par le biais de l’un de ses deux cas spécialisés.

Figure 1-20. *Relation de généralisation entre cas d’utilisation*

Le diagramme des cas d’utilisation du client banque fait apparaître également la fac- torisation rendue possible de la relation d’inclusion avec le fragment de cas d’utilisa- tion *S’authentifier*. Il permet surtout d’associer l’acteur secondaire S.I. Banque avec le seul cas d’utilisation spécialisé *Déposer du numéraire*.

296723AWP-UML2\_8E.book Page 50 Lundi, 19. février 2018 7:54 07

**Point 50**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 1-21. *Cas d’utilisation du client banque*

Avec tous ces ajouts, que devient donc notre diagramme de cas d’utilisation ? Il est maintenant si complexe (comparé à la figure 1-7) qu’il serait illusoire de penser qu’il puisse être lisible en une seule page, comme le montre le schéma suivant.

Notez que nous avons introduit un acteur généralisé abstrait « *Porteur* » en tant qu’acteur principal du fragment de cas d’utilisation s’authentifier. Cela nous permet ainsi d’inclure ce fragment dans les cas d’utilisation du client mais aussi du porteur non client.

Notez également que nous avons distingué les cas d’utilisation de l’opérateur de main- tenance par le mot-clé « support », pour mettre l’accent sur la différence de niveau avec les cas principaux que sont ceux des porteurs de carte. Ils sont en général moins importants fonctionnellement, mais ne doivent cependant pas être oubliés. Imaginez un GAB que l’on ne pourrait pas recharger en billet !

296723AWP-UML2\_8E.book Page 51 Lundi, 19. février 2018 7:54 07

Figure 1-22. *Diagramme de cas d’utilisation complet du GAB*

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **51**

Pour améliorer notre modèle, nous allons donc organiser les cas d’utilisation et les regrouper en ensembles cohérents. Pour ce faire, nous utilisons le mécanisme général de regroupement d’éléments en UML, qui s’appelle le *package*. Le package est une sorte de dossier permettant de structurer un modèle en unités cohérentes. Les outils de modélisa- tion du marché se servent pour la plupart de ce concept comme unité de gestion de version, de stockage, et de partage du modèle pour le travail en équipe. Nous y revien- drons plus en détail dans la partie II (modélisation statique).

**EXERCICE 1-14. Structuration des cas d’utilisation en packages**

**Proposez une structuration des cas d’utilisation du GAB en packages.**

Plusieurs stratégies sont possibles : procéder au regroupement par acteur, par domaine fonctionnel, etc. Dans notre exemple, un regroupement des cas d’utilisation par acteur principal s’impose, car cela permet également de répartir les acteurs secondaires.

296723AWP-UML2\_8E.book Page 52 Lundi, 19. février 2018 7:54 07

**Point 52**

**de vue fonctionnel** *PREMIÈRE PARTIE* Le cas d’utilisation inclus *S*’*authentifier* est mis dans un package à part, en tant que fragment commun, pour bien le distinguer des vrais cas fonctionnels qui l’incluent. Les flèches de dépendance entre packages de cas d’utilisation synthétisent les éven- tuelles relations entre les cas, c’est-à-dire ici les inclusions. Le schéma suivant pré- sente la structuration proposée des cas d’utilisation. Il s’agit d’un diagramme de packages, officialisé par UML 2.

Figure 1-23. *Diagramme de packages des cas d*’*utilisation du GAB*

Il est maintenant possible de dessiner un diagramme de cas d’utilisation par pac- kages. Cela ne présente aucune difficulté et nous donnerons uniquement le dia- gramme du premier package, pour montrer comment apparaît le fragment appartenant à un package différent. La notation (*from Fragments*) n’est pas standard UML, mais est utilisée par plusieurs outils du marché.

Figure 1-24. *Diagramme de cas d’utilisation du package* Opérations non client

296723AWP-UML2\_8E.book Page 53 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **53**

**Étape 7 – Dynamique globale : Interaction Overview Diagram**

Dans cette dernière étape, nous allons explorer l’utilisation d’un nouveau type de diagramme UML 2 : *Interaction Overview Diagram*13. Ce diagramme est une fusion intéressante du diagramme d’activité et du diagramme de séquence ! Il permet d’organiser des interactions, représentées par exemple par des diagrammes de séquence, au moyen des nœuds de contrôle du diagramme d’activité : décision, parallélisme, etc. C’est donc une sorte de diagramme d’activité14 dans lequel les actions sont remplacées par des interactions.

.

Représentons d’abord le processus d’authentification, suivi du choix du type de transaction : retrait, consultation ou dépôt. Le diagramme global est dessiné sur la figure 1-25.

13. Nous préférons conserver le nom d’origine des nouveaux types de diagrammes UML 2 faute d’une traduction française

évidente (diagramme de vue globale d’interaction ? diagramme de vue d’ensemble des interactions ? etc.). 14. Pourtant, l’outil utilise le tag sd, car dans les spécifications UML 2, ce diagramme est classé dans la catégorie des dia- grammes d’interaction, et ceux-ci (séquence, communication, timing, interaction overview) sont tous représentés avec un tag sd.

**EXERCICE 1-15.** *Interaction Overview Diagram*

**Représentez la dynamique globale du GAB dans le point de vue de l’acteur** Client banque**, en représentant les interactions de ses cas d’utilisation dans un** *Interaction Overview Diagram***. Modélisez en particulier le fait que le client peut enchaîner plu- sieurs transactions (retrait, dépôt, etc.) sans avoir à s’authentifier de nouveau. Compa- rez avec un diagramme de séquence équivalent.**

296723AWP-UML2\_8E.book Page 54 Lundi, 19. février 2018 7:54 07

**Point 54**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 1-25. *Vue d’ensemble des interactions du client banque*

Il est également possible de remplacer chaque référence par un diagramme de séquence « inline ». Nous l’avons fait pour l’interaction S’authentifier. Il est clair que le souci de lisibilité du diagramme empêche de remplacer chaque référence par un diagramme de séquence entier.

296723AWP-UML2\_8E.book Page 55 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : étude de cas**

*CHAPITRE 1* **55**

Figure 1-26. *Vue d’ensemble des interactions du client banque (expansé)*

296723AWP-UML2\_8E.book Page 56 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **56**

Essayons maintenant de présenter un diagramme de séquence équivalent. Nous devons remplacer le choix du type de transaction par un fragment combiné alt, et l’inclure dans un fragment loop. On utilise les mêmes cadres ref que dans le diagramme de vue globale pour référencer les morceaux d’interaction. Dans cet exemple précis, la valeur ajoutée de l’*Interaction Overview Diagram* est loin d’être évidente...

Figure 1-27. *Diagramme de séquence du comportement du client banque*

296723AWP-UML2\_8E.book Page 57 Lundi, 19. février 2018 7:54 07

**2**

Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques

**Mots-clés**

■ Diagramme de cas d’utilisation ■ Relations entre cas d’utilisation ■ Navigabilité des associa- tions 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 ■ Use case vs. User story

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 navi- gabilité 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.

296723AWP-UML2\_8E.book Page 58 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **58**

**Étude d’un terminal point de vente**

Cet exercice concerne un système simplifié de caisse enregistreuse de super- marché. Il est largement inspiré de l’étude de cas initialement proposée par [Larman 97].

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 ;

296723AWP-UML2\_8E.book Page 59 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **59**

– 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 informa- tions 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.

**Étape 1 – Réalisation du diagramme de cas d’utilisation**

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 respon- sable du magasin. Le dessin est souvent fait à la main, par exemple sur un tableau blanc en réunion, pour démarrer la discussion.

Figure 2-1. *Première ébauche 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 dia- gramme plus précis.**

296723AWP-UML2\_8E.book Page 60 Lundi, 19. février 2018 7:54 07

**Point 60**

**de vue fonctionnel** *PREMIÈRE PARTIE* 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 séquentielle- ment comme cela est illustré sur la figure 2-3.

296723AWP-UML2\_8E.book Page 61 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **61**

Figure 2-3. *Découpe séquentielle du cas d’utilisation principal*

Quoique tentante, cette solution est rarement recommandable. En effet, les cas d’uti- lisation 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 *opé- rations système*1, 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 procé- dure 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 2-4.

1. 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 ».

296723AWP-UML2\_8E.book Page 62 Lundi, 19. février 2018 7:54 07

**Point 62**

**de vue fonctionnel** *PREMIÈRE PARTIE* Figure 2-4. *Diagramme de cas d’utilisation partiel*

Le cas d’utilisation inclus, *Traiter le paiement*, est porté en italique sur le dia- gramme 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’utilisa- tion 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.

296723AWP-UML2\_8E.book Page 63 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **63**

Figure 2-5. *Diagramme de cas d’utilisation complété***Étape 2 – Descriptions essentielle et réelle d’un cas d’utilisation**

**À RETENIR**

Nous allons illustrer cette différence avec les deux questions suivantes.

**EXERCICE 2-2. Description essentielle d’un cas d’utilisation**

Cas d'utilisation essentiel/réel

C. Larman a introduit dans [Larman 97] la distinction entre cas d’utilisation essentiel et cas d’uti- lisation 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.

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

296723AWP-UML2\_8E.book Page 64 Lundi, 19. février 2018 7:54 07

**Point 64**

**de vue fonctionnel** *PREMIÈRE PARTIE* **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** : 17/10/16 **Date de mise à jour** : 11/05/17 **Version** : 1.8 **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, exé- cuter 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.

296723AWP-UML2\_8E.book Page 65 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **65**

**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** : 17/10/17 **Date de mise à jour** : 11/05/17 **Version** : 1.8 **Responsable** : Pascal Roques

**Description des scénarios**

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

296723AWP-UML2\_8E.book Page 66 Lundi, 19. février 2018 7:54 07

**Point 66**

**de vue fonctionnel** *PREMIÈRE PARTIE* **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 éventuel- lement plus élevée que le montant total de la vente. 3. Le Caissier enregistre la somme don-

née par le 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. 4. 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 paie- ment 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 ren-

dre 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.

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.

296723AWP-UML2\_8E.book Page 67 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **67**

**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.**

Le sommaire d’identification est similaire au précédent, mais le type devient : réel détaillé. L’interface homme-machine propo- sée est dessinée sur la figure sui- vante. Nous avons utilisé pour cela l’excellent outil de maquettage Balsamiq Mockups, très à la mode dans la communauté agile (http:// www.balsamiq.com).

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 univer- sel d’identification du produit dans le champ CPU de la fenêtre de dialo- gue 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 vali- dation : « 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 ».

296723AWP-UML2\_8E.book Page 68 Lundi, 19. février 2018 7:54 07

**Point 68**

**de vue fonctionnel** *PREMIÈRE PARTIE* 6. Le Caissier annonce le montant total

au Client.

7. Le Client choisit le type de paiement :

• En cas de paiement cash, exécuter le cas d’utilisation « Traiter le paie- ment en liquide ».

• En cas de paiement par carte de cré- dit, exécuter le cas d’utilisation « Traiter le paiement par carte de crédit ».

• En cas de paiement par chèque, exé- cuter 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 éventuel- lement plus élevée que le montant total de la vente. 3. Le Caissier enregistre la somme don- née par le client dans le champ « Paiement ». Il valide au moyen du bouton « Sai- sie 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.

296723AWP-UML2\_8E.book Page 69 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **69**

**AVIS D’EXPERT**

Jean-Claude Grosjean (ergonome consultant, coach agile ; http://www.qualitystreet.fr/)

*Cas d’utilisation UML ... oui mais ... (5 octobre 2007)*

N’oubliez pas qu’un cas d’utilisation est avant tout textuel, et n’associez donc pas aussi radicalement ce cas d’utilisation (use case) au diagramme UML : privilégiez plutôt la démarche. En effet, se lancer dans la rédaction des cas d’utilisation, pour décrire un besoin fonction- nel (spécifications), c’est se lancer dans une véritable démarche d’analyse, progressive, parfois lente, parsemée d’ateliers de travail, d’entretiens ... C’est aussi adopter une vraie réflexion en termes d’utilisateurs (acteurs), de buts et de tâches. Croyez-moi, c’est bien là l’essentiel. Le diagramme des cas d’utilisation UML (use case diagram) est, quant à lui, très précieux pour bénéficier d’une vue globale sur l’application ; il permet de visualiser immédiatement les liens entre acteurs et cas d’utilisation, ou encore de délimiter explicitement les différents packages. « Modéliser graphiquement » est un principe du processus unifié (mais toujours fonction des destinataires), donc le diagramme des use cases ne doit absolument pas être négligé ! Pour ma part, ce n’est pourtant pas le diagramme qui m’a séduit... J’ai découvert les cas d’utilisation en 2000, j’étais alors consultant au Luxembourg, et j’ai très rapidement perçu, en les construisant (et grâce à de bons mentors), la forte complé- mentarité à la fois avec la démarche ergonomique (profils utilisateurs, réflexion sur les buts et scénarios, UCD...) et avec les activités, livrables de l’ergonome ou designer d’interaction (personas, storyboard, diagramme de tâches, wireframes). Le fait que les cas d’utilisation se focalisent seulement sur le Quoi (fonctionnel et métier) – c’est une règle d’or –, sans décrire les éléments d’interfaces (écrans et enchaînement), laissés aux spécialistes de l’IHM, est aussi un élément que j’ai beaucoup apprécié, selon moi un vrai point fort. Donc, depuis tout ce temps, j’évangélise... en insistant principalement sur 6 points :

• La démarche de découverte et de construction progressive des cas d’utilisation : l’essentiel...

• La forte adéquation avec le développement itératif (dans l’estimation, la priorisation, la planification, le traitement).

• La complémentarité avec le travail de l’ergonome.

• La lisibilité, le formalisme des cas d’utilisation (élément clé de son efficacité et de son acceptation par les équipes).

• La gestion des modifications (pas si simple que ça !).

• Le lien fort avec les cas de tests et une approche de validation fonctionnelle (c’est l’idéal !). Et je recommande toujours l’ouvrage d’Alistair Cockburn : *Rédiger des cas d’utilisation effi- caces* (aux éditions Eyrolles), la référence que je conseille à tous ceux qui souhaitent s’attaquer à l’analyse système ou métier. Enfin, même si aujourd’hui je me retrouve davantage dans les user stories, je reste convaincu de la pertinence des cas d’utilisation dans pas mal de contextes... quand ils sont bien rédigés !

296723AWP-UML2\_8E.book Page 70 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **70**

**Étape 3 – Description graphique des cas d’utilisation**

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

**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.**

296723AWP-UML2\_8E.book Page 71 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **71**

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 néan- moins 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 vrai- ment 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 trai- tements du système, et la description des pré- et postconditions 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*

TPV *de* Traiter le passage en caisse

296723AWP-UML2\_8E.book Page 72 Lundi, 19. février 2018 7:54 07

**Point de vue fonctionnel** *PREMIÈRE PARTIE* **72**

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

: TPV

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 diffé- rents types de paiement. Proposez également un diagramme supplémentaire montrant le travail du caissier pendant les heures d’ouverture du magasin.**

296723AWP-UML2\_8E.book Page 73 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **73**

Pour prendre en compte les différents types de paiement, on peut remplacer le message saisieMontantPayé(montant) du diagramme précédent par une référence vers une inte- raction concernant le cas d’utilisation inclus *Traiter le paiement*.

:TPV

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*).

296723AWP-UML2\_8E.book Page 74 Lundi, 19. février 2018 7:54 07

**Point 74**

**de vue fonctionnel** *PREMIÈRE PARTIE* 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 rela- tive 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*

296723AWP-UML2\_8E.book Page 75 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **75**

Le parallélisme peut être représenté en UML 2 par un fragment combiné par conte- nant deux opérandes. Dans le second, il suffit d’encadrer la production du ticket par un fragment optionnel, contenant lui-même un deuxième fragment pour la récupération optionnelle du ticket, comme illustré par la figure 2-12.

Figure 2-12. *Diagramme de séquence amélioré de la fin du retrait d’argent*

**Étape 4 – Réalisation d’un diagramme d’états au niveau système**

**EXERCICE 2-6. Diagramme de séquence système (fin)**

**Reprenez le diagramme de séquence 1-13 du chapitre précédent. Proposez une fin différente utilisant un fragment optionnel (**opt**) pour la production du ticket. Représentez également le fait que les billets et le ticket éventuel sont sortis en même temps par le GAB.**

**EXERCICE 2-7. 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 paie- ment cash. Étendez ensuite le diagramme en prenant en compte les différents types de paiement, ainsi que les autres actions du caissier.**

296723AWP-UML2\_8E.book Page 76 Lundi, 19. février 2018 7:54 07

**Point 76**

**de vue fonctionnel** *PREMIÈRE PARTIE* Les opérations système identifiées grâce à l’exercice 2-4 correspondent aux trois

messages2 dont le système est destinataire :

Figure 2-13. *Opérations système de* Traiter le passage en caisse

Pour représenter la séquence forcée de ces trois opérations système, avec la répétition possible de la saisie d’article, un diagramme d’états s’impose3. Il représente en fait le sous-ensemble des états de la caisse induits par le cas d’utilisation *Traiter le passage en caisse*.

Figure 2-14. *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 légè- rement plus complexe.

2. 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 dia- gramme d’états 2.14. 3. 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.

296723AWP-UML2\_8E.book Page 77 Lundi, 19. février 2018 7:54 07

**Modélisation fonctionnelle : exercices corrigés et conseils méthodologiques**

*CHAPITRE 2* **77**

Figure 2-15. *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 2-14) :

Figure 2-16. *Diagramme d’états de la caisse (paiement cash)*

saisieArticle

Libre saisieArticle

Enregistrement des articles

autorisation[ OK ]

saisiePaiement[ liquide ] finDeVente Attente autorisation

saisiePaiement[chèque ou CB ]

Attente paiement

autorisation[ non OK ]