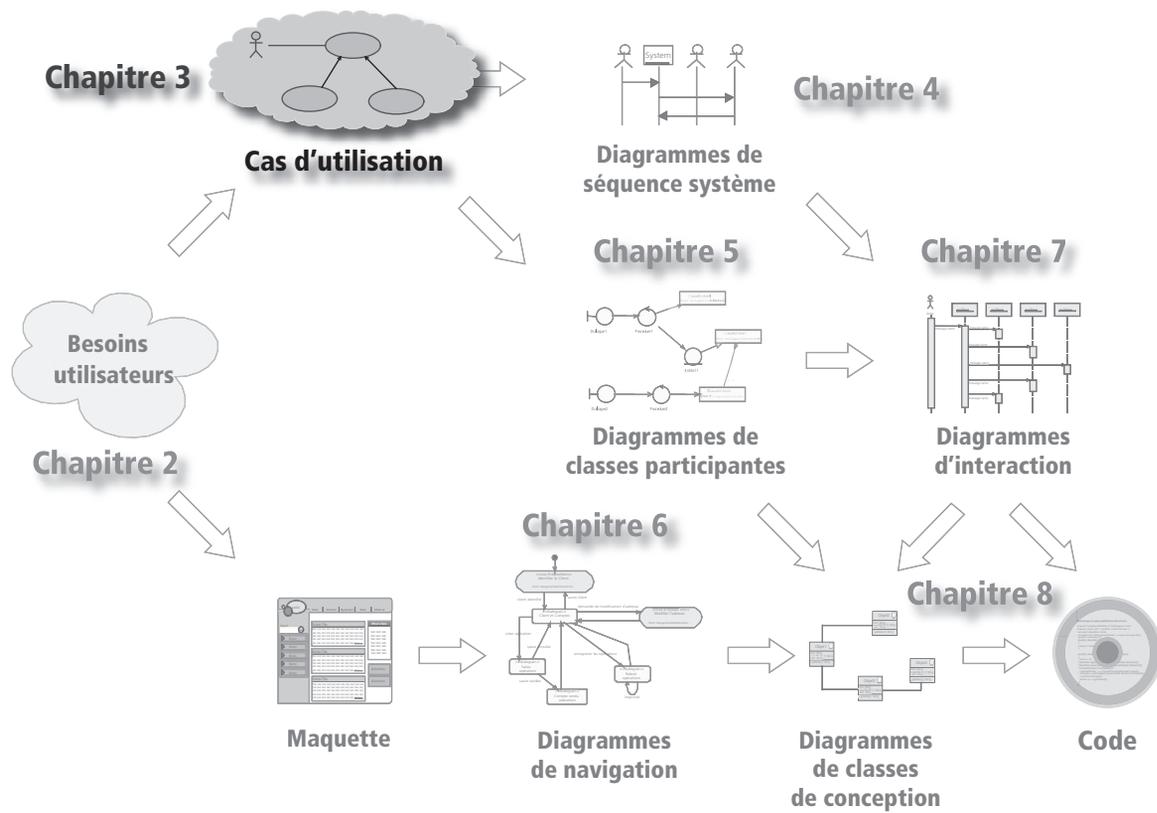


chapitre 3



Spécification des exigences d'après les cas d'utilisation

Acteurs et cas d'utilisation sont les concepts UML fondamentaux pour la spécification des exigences. Nous apprendrons à les identifier à partir de l'expression initiale des besoins de notre étude de cas. Nous verrons ensuite comment structurer, relier et classer ces cas d'utilisation ainsi que les représentations graphiques UML associées. Nous aborderons enfin l'impact de cette étude sur la planification du projet et les bénéfices qu'en tire le chef de projet, ainsi que la problématique de la traçabilité avec les exigences textuelles.

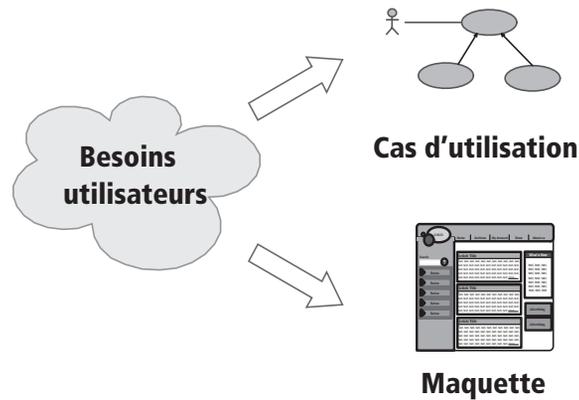
SOMMAIRE

- ▶ Identification des acteurs
- ▶ Identification des cas d'utilisation
- ▶ Structuration en packages
- ▶ Relations entre cas d'utilisation
- ▶ Classement des cas d'utilisation
- ▶ Planification du projet en itérations
- ▶ Traçabilité avec les exigences

MOTS-CLÉS

- ▶ Acteur
- ▶ Cas d'utilisation
- ▶ Package
- ▶ Planification
- ▶ Itération
- ▶ Traçabilité

Figure 3-1
Les besoins donnent lieu à des cas d'utilisation et à une maquette.



Démarche

Rappelons comment cette activité de spécification des exigences se situe par rapport à l'ensemble du processus décrit au chapitre 1. L'expression préliminaire des besoins donne lieu à une modélisation par les cas d'utilisation et à une maquette d'interface homme-machine (IHM), comme indiqué sur la figure 3-1.

Dans ce chapitre, nous allons détailler la branche supérieure concernant la modélisation des cas d'utilisation. La réalisation d'une maquette graphique est une activité courante mettant en jeu des outils de dessin. Elle montre rapidement l'aspect visuel (le « look ») du site web. De nombreux ouvrages traitent déjà de ce sujet particulier qui est en dehors de la portée de ce livre.

Nous allons au contraire nous concentrer sur ce qu'il y a « derrière » la maquette, c'est-à-dire : quelles sont les informations à montrer, à qui et pour quoi faire ? Quelles sont les informations à montrer ? C'est ce que nous décrirons dans le chapitre 5 traitant des classes d'analyse. À qui et pour quoi faire ? C'est précisément ici que les concepts UML d'acteurs et de cas d'utilisation interviennent.

Détaillons un peu plus les différentes étapes de la démarche que nous allons adopter afin d'aboutir au modèle des cas d'utilisation (voir la figure 3-2) :

- identifier les acteurs,
- identifier les cas d'utilisation,
- structurer les cas d'utilisation en packages,
- ajouter les relations entre cas d'utilisation,
- finaliser un ou plusieurs diagramme(s) de cas d'utilisation par package.

REFFÉRENCES Design graphique de site web

- 📖 *Initiation à la conception Web : Cours et exercices*, J. Sklar, Eyrolles, 2001
- 📖 *Réussir un projet de site web*, N. Chu, Eyrolles, 2006
- 📖 *Sites web – Les bonnes pratiques*, E. Sloïm, Eyrolles 2007
- 📖 *CSS 2 – Pratique du design web*, R. Goetter, Eyrolles 2007
- 📖 *Réussir son site web avec XHTML et CSS*, M. Nebra, Eyrolles 2006
- 📖 *Je crée mon site Internet avec Dreamweaver 8 et Flash 8*, C. Bergé, Eyrolles 2006
- 📖 *Design web : utiliser les standards – CSS et XHTML*, J. Zeldman, Eyrolles 2006
- 📖 *Création et optimisation de sites web – CSS2, RSS, XHTML et XML*, Collectif Campus Press, 2006
- 📖 *CSS 2 – Pratique du design web*, R. Goetter, First Eyrolles, 2005
- 📖 *Ergonomie du logiciel et design web*, J.F. Nogier, Dunod, 2005

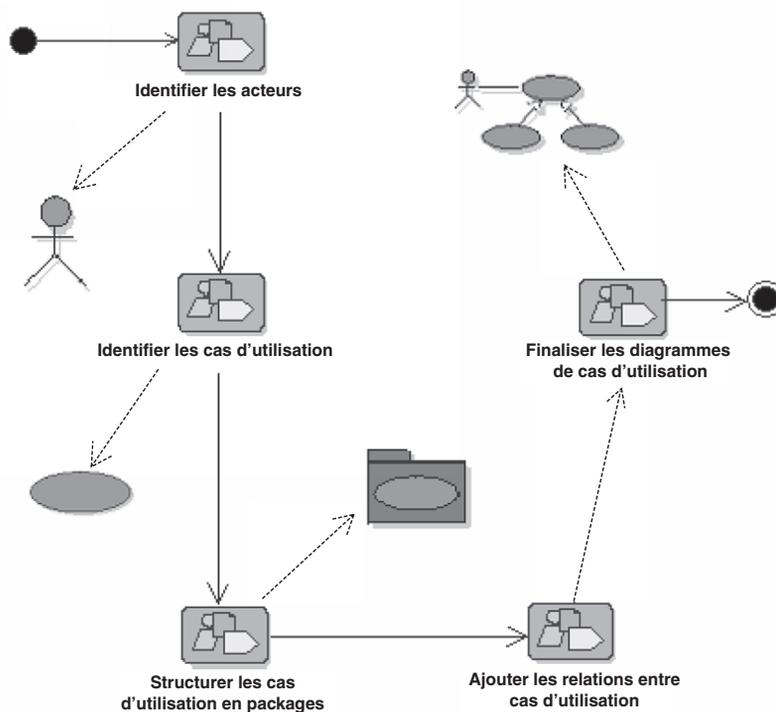


Figure 3-2
Synoptique de la démarche de construction
du modèle des cas d'utilisation

Le modèle ainsi obtenu permet d'établir les priorités entre les cas d'utilisation afin d'aider le chef de projet à planifier ses itérations en connaissance de cause.

Identification des acteurs

Les acteurs humains pour le site web www.jeBouquine.com sont les suivants :

- **L'Internaute** : la personne qui visite le site pour rechercher des ouvrages et éventuellement passer une commande. Il s'agit bien sûr de l'acteur le plus important, celui pour lequel le site existe !
- **Le Webmaster** : rôle des employés qui sont en charge du bon fonctionnement et de la maintenance du site web.
- **Le Service clients** : rôle des employés qui s'occupent du suivi des commandes des clients.
- **Le Libraire** : rôle des employés qui sont responsables du contenu rédactionnel du site.

Nous allons également prendre en compte les systèmes informatiques connectés au site web, à savoir le système Nouveautés qui alimente la base avec tous les nouveaux ouvrages, et le système Gestion des stocks

B.A.-BA 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.

ATTENTION

Acteurs logiques vs acteurs physiques

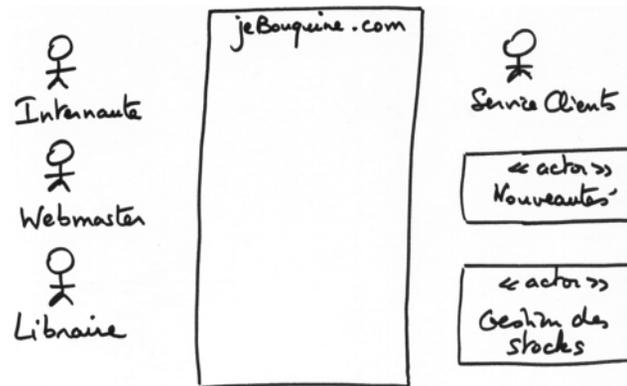
Éliminez autant que faire se peut les acteurs « physiques » au profit des acteurs « logiques » : l'acteur est celui qui bénéficie de l'utilisation du système. Cette règle permet en particulier de s'affranchir des technologies d'interface qui risquent d'évoluer au fil du projet.

ATTENTION Rôle vs entité concrète

Ne confondez pas rôle et entité concrète. Une même entité concrète peut jouer successivement différents rôles par rapport au système étudié, et être modélisée par plusieurs acteurs. Réciproquement, le même rôle peut être tenu simultanément par plusieurs entités concrètes, qui seront alors modélisées par le même acteur. Par conséquent, même si une seule personne physique peut jouer successivement les rôles de Libraire et Webmaster vis à vis du site web, il s'agit bien de deux acteurs distincts, de deux profils différents.

Figure 3-3

Acteurs du site jeBouquine.com



Identification des cas d'utilisation

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions « métier » selon lesquelles il utilise le système.

Commençons par l'acteur le plus important pour un site d'e-commerce : l'Internaute.

ATTENTION Cas d'utilisation = ensemble de séquences d'actions

Une erreur fréquente concernant les cas d'utilisation consiste à vouloir descendre trop bas en termes de granularité. Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système, et le lien entre ces séquences d'actions est précisément l'objectif métier de l'acteur. Le cas d'utilisation ne doit donc pas se réduire systématiquement à une seule séquence, et encore moins à une simple action.

B.A.-BA 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.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné.

Ces cas d'utilisation principaux ont été bien mis en évidence par l'expression de besoins préliminaire du chapitre précédent, à savoir :

- rechercher des ouvrages,
- gérer son panier,
- effectuer une commande.

On obtient un diagramme préliminaire (voir figure 3-4) en représentant sur un schéma les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs. Un cas d'utilisation doit être relié à au moins un acteur.

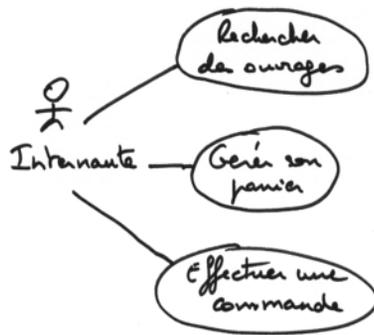


Figure 3-4
Cas d'utilisation principaux de l'internaute

L'internaute peut également consulter l'historique de ses commandes en cours.

Il faut bien comprendre que chaque cas d'utilisation doit avoir un objectif en soi et pouvoir être réalisé indépendamment des autres. Un internaute visite quelquefois le site dans le seul but de chercher des ouvrages, sans intention d'acheter. Dans d'autres cas, il gère un panier virtuel pour faire une simulation, ou obtenir un devis.

Il peut également se connecter pour surveiller l'état de sa dernière commande. Tous ces objectifs sont bien indépendants et auto-suffisants : il s'agit de vrais cas d'utilisation ! L'ensemble des cas d'utilisation de l'internaute est représenté sur la figure 3-5.

CONCEPT AVANCÉ Flèche sur l'association

L'utilisation de la flèche sur l'association entre le cas d'utilisation Effectuer une commande et l'acteur Service clients signale un sens unique de transmission d'information. Cet acteur ne fait que recevoir des messages du système, sans lui en envoyer, dans le cadre de ce cas d'utilisation. En revanche, dans le cadre du cas d'utilisation Consulter ses commandes, l'acteur Service clients va interagir dans les deux sens avec le système. Remarquez également qu'un cas d'utilisation peut faire participer plusieurs acteurs et qu'un acteur peut participer à plusieurs cas d'utilisation.

MÉTHODE Nom des cas d'utilisation

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).

B.A.-BA Association

Une association est une relation entre éléments UML (classes, cas d'utilisation, etc.) qui décrit un ensemble de liens. Elle est utilisée dans le cadre du diagramme de cas d'utilisation pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement : « participe à ».

ATTENTION Taille des cas d'utilisation

Obtenir un devis est trop « petit » pour être un cas d'utilisation ! Rappelez-vous qu'un cas d'utilisation est un ensemble de séquences d'actions : sûrement pas une seule action.

B.A.-BA Acteur principal

Contrairement à ce que l'on pourrait croire, tous les acteurs n'utilisent pas forcément le système ! 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 complémentaires ; ils peuvent uniquement consulter ou informer le système lors de l'exécution du cas d'utilisation. Une bonne pratique consiste à faire figurer les acteurs principaux à gauche des cas d'utilisation et les acteurs secondaires à droite.

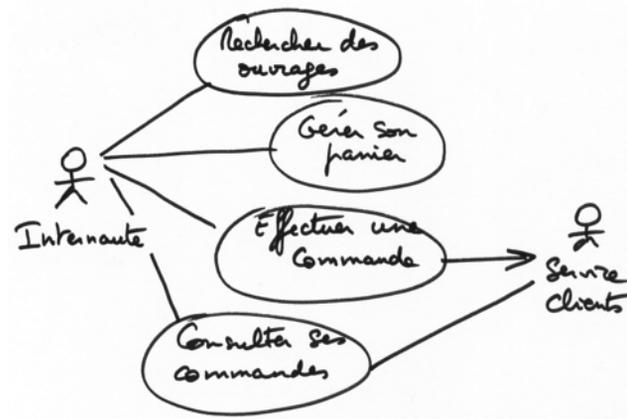


Figure 3-5
Cas d'utilisation de l'internaute

CONCEPT AVANCÉ Flèche sur l'association

Notez cette fois-ci les flèches de navigabilité vers le cas d'utilisation Maintenir le catalogue : les acteurs non humains ne font qu'envoyer des messages au système, sans jamais en recevoir. Ce sont de parfaits exemples d'acteurs secondaires.

Quels sont les cas d'utilisation des employés de jeBouquine ? D'après le chapitre 2, on identifie :

- maintenir le catalogue, qui fait intervenir les deux systèmes Nouveautés et Gestion des stocks,
- maintenir les informations éditoriales,
- maintenir le site.

Représentons ces cas sur un diagramme avec leurs acteurs associés (voir figure 3-6).

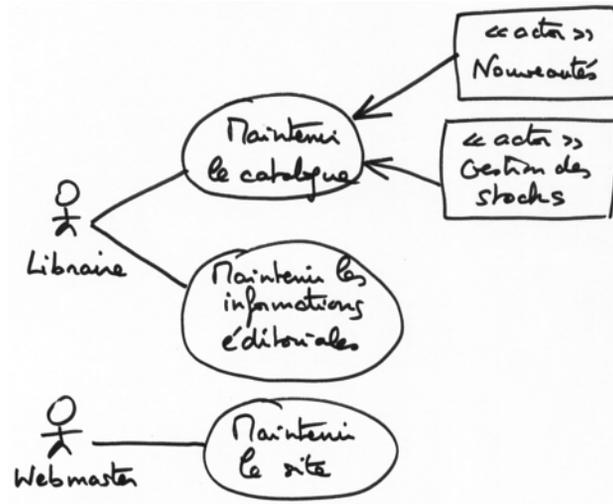


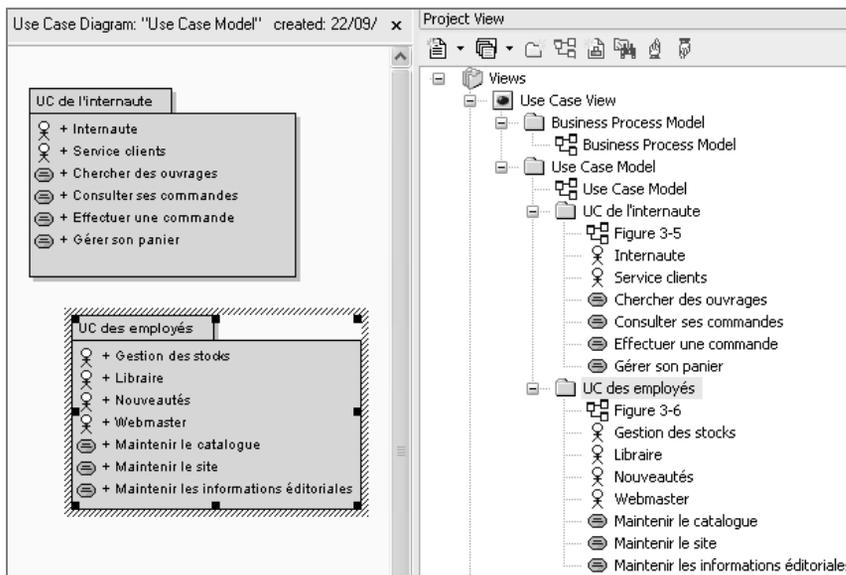
Figure 3-6
Cas d'utilisation des employés

Structuration en packages

Pour améliorer notre modèle, nous allons organiser les cas d'utilisation et les regrouper en ensembles fonctionnels cohérents. Pour ce faire, nous utilisons le concept général d'UML, le package.

Les acteurs ont également été regroupés dans un package séparé sur lequel s'appuient les deux packages de cas d'utilisation. L'organisation générale du modèle dans un outil de modélisation est représentée sur la figure 3-7.

Le sigle UC pour use case est souvent utilisé pour raccourcir les noms de packages.



B.A.-BA Package

Le package est un mécanisme général de regroupement d'éléments en UML, qui peut être utilisé par exemple pour regrouper des cas d'utilisation, mais aussi des acteurs, des classes, etc.

OUTIL Enterprise Architect

L'outil UML 2 que nous avons utilisé dans cet ouvrage s'appelle Enterprise Architect, de la société Sparx Systems. Un petit clin d'œil à cette firme australienne qui nous a aimablement fourni une licence gratuite complète.

► www.sparxsystems.com

Notez également que nos premiers diagrammes ont souvent été faits « à la main », afin de montrer qu'il n'est pas indispensable d'acheter un progiciel coûteux pour utiliser UML !

Figure 3-7
Organisation des cas d'utilisation et des acteurs en packages (avec l'outil Enterprise Architect)

Affinement du modèle de cas d'utilisation

Après une nouvelle réunion d'expression des besoins avec le responsable du projet, nous sommes arrivés à la conclusion qu'il était nécessaire de distinguer deux profils d'internautes :

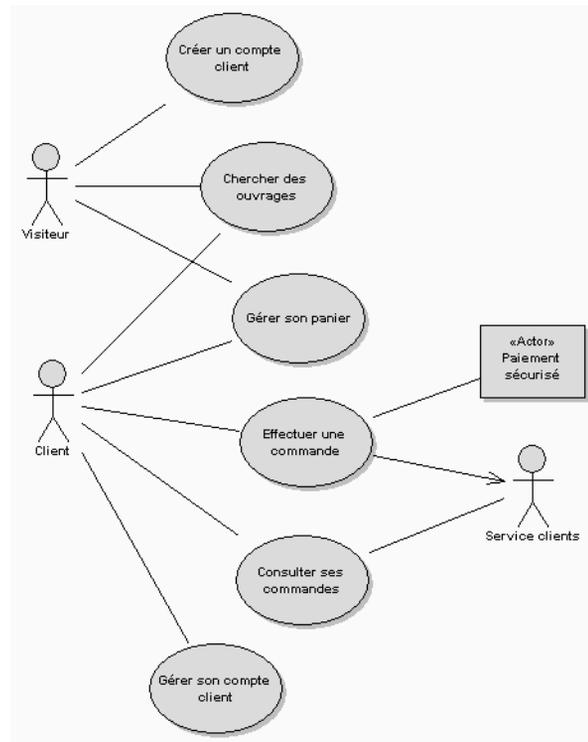
- le **Visiteur**, inconnu du site web, mais qui peut néanmoins rechercher des ouvrages et gérer un panier ;
- le **Client**, déjà connu par le site web, qui peut seul effectuer une commande et en suivre l'état.

À tout moment, le visiteur peut choisir de créer un compte, afin de devenir client. Le client a également la possibilité de modifier les informations le concernant (adresse de facturation, adresses de livraison, adresse électronique, etc.) stockées par le site web.

Nous avons également oublié de mentionner l'important système externe de Paiement sécurisé, nécessaire au moment du paiement en ligne.

Le diagramme de cas d'utilisation des internautes devient donc tel que représenté sur la figure 3-8.

Figure 3-8
Cas d'utilisation
du Visiteur et du Client



B.A.-BA Acteur généralisé

Il arrive que deux acteurs, ou plus, présentent des similitudes dans leurs relations aux cas d'utilisation. On peut l'exprimer en créant un acteur généralisé, éventuellement abstrait, qui modélise les aspects communs aux différents acteurs concrets. Dans notre exemple, l'acteur Internaute est la généralisation abstraite des rôles Visiteur et Client. Notez que le nom d'un acteur abstrait s'écrit en italique.

Nous remarquons maintenant que les deux acteurs Client et Visiteur partagent deux cas d'utilisation : ils sont également acteurs principaux de Chercher des ouvrages et Gérer son panier. Or, une bonne pratique consiste à identifier un seul acteur principal par cas d'utilisation. UML nous fournit la solution en permettant de créer un acteur généralisé Internaute, dont Client et Visiteur seront des spécialisations.

Le diagramme devient alors celui de la figure 3-9, avec un seul acteur principal par cas d'utilisation.

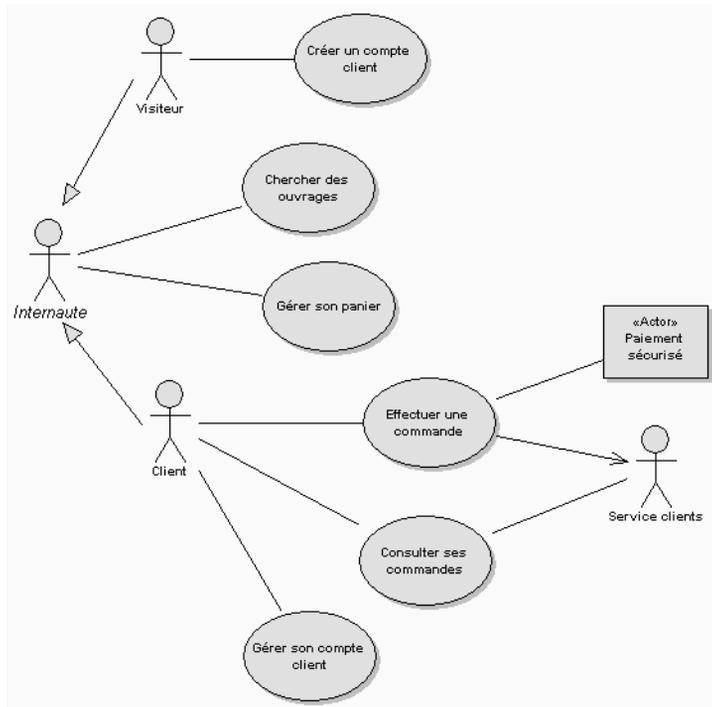


Figure 3-9
Cas d'utilisation des internautes

On pourrait relier les cas d'utilisation des internautes par des relations d'extension :

- La recherche d'ouvrages peut donner lieu à leur mise dans le panier (et réciproquement !).
- La gestion du panier peut donner lieu au passage d'une commande.
- Le passage d'une commande peut donner lieu à la gestion du compte client (ajout d'une adresse, etc).

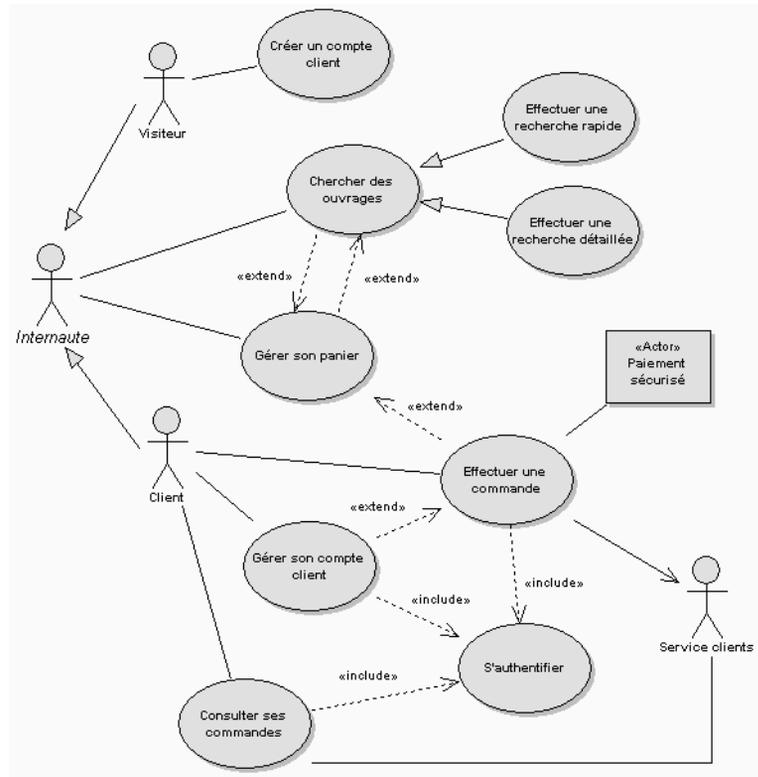
De même, les différentes possibilités de recherche d'ouvrages seront modélisées plus finement par une relation de généralisation/spécialisation.

Enfin, l'authentification du client est nécessaire au début du passage d'une commande, du suivi des commandes, ou de la modification des informations du compte.

Toutes ces relations entre cas d'utilisation, légales en UML, mais souvent mal utilisées et sources de confusion pour les experts métier, sont illustrées sur la figure 3-10.

Pensez-vous vraiment que le diagramme ainsi obtenu soit satisfaisant ? Non, même s'il est légal en UML, ce schéma est devenu maintenant trop complexe par rapport à l'intérêt de l'information ajoutée. Nous vous mettons formellement en garde contre l'usage abusif des relations entre cas d'utilisation !

Figure 3-10
Relations entre
cas d'utilisation des internautes



**ATTENTION N'abusez pas
des relations entre cas d'utilisation !**

N'abusez pas des relations entre cas d'utilisation (inclusion, extension, généralisation) : elles peuvent rendre les diagrammes de cas d'utilisation trop difficiles à décrypter pour les experts métier qui sont censés les valider.

CONCEPT AVANCÉ Relations entre cas d'utilisation

Pour affiner le diagramme de cas d'utilisation, UML définit trois types de relations standardisées entre cas d'utilisation :

- Une relation d'inclusion, formalisée par le mot-clé `<<include>>` : le cas d'utilisation de base incorpore explicitement un autre, de façon obligatoire.
- Une relation d'extension, formalisée par le mot-clé `<<extend>>` : le cas d'utilisation de base incorpore implicitement un autre, de façon optionnelle.
- Une relation de généralisation/spécialisation : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.

CONCEPT AVANCÉ Types de relations

Notez les trois différents types de relations présentes sur le diagramme 3-10 :

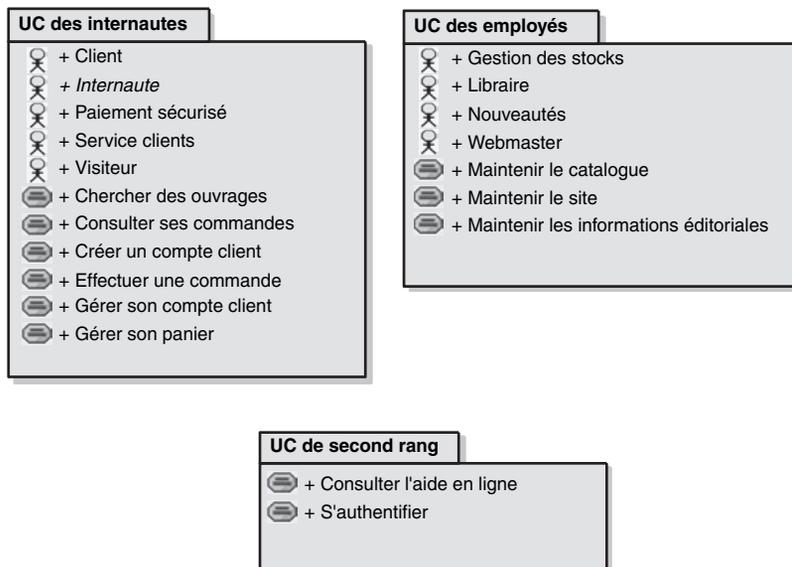
- l'association (trait plein avec ou sans flèche, comme sur la figure 3-8) entre acteurs et cas d'utilisation ;
- la dépendance (flèche pointillée) entre cas d'utilisation, avec les mots-clés `<<extend>>` ou `<<include>>` ;
- la relation de généralisation (flèche fermée vide) entre cas d'utilisation.

Trop de projets en ont souffert, avec une incompréhension légitime de la part des experts métier à qui l'on demande de valider ces diagrammes. La figure 3-9 nous paraît donc largement suffisante pour exprimer les besoins fonctionnels majeurs de l'application web.

Néanmoins, le cas d'utilisation *S'authentifier* est à conserver, puisqu'il devra être réalisé afin de permettre au *Client* d'exécuter ses propres cas d'utilisation majeurs. Nous qualifierons donc ce petit cas d'utilisation de « fragment » : il ne représente pas un objectif à part entière du client, mais plutôt un objectif de niveau intermédiaire.

Avons-nous maintenant identifié tous les cas d'utilisation ? En fait, non. Nous avons omis un cas d'utilisation un peu particulier de l'internaute : *Consulter l'aide en ligne*. Certes, il ne s'agit pas non plus d'un cas d'utilisation majeur, mais il donnera lieu à des développements importants au niveau du projet informatique ! Il ne faut donc pas l'oublier, même si nous le qualifions de « secondaire » par rapport aux autres cas identifiés précédemment.

Pour bien structurer notre expression de besoin, nous préférons isoler les cas d'utilisation de second rang (*S'authentifier* et *Consulter l'aide en ligne*) dans un package à part, intitulé UC de second rang. L'organisation du modèle devient alors celle indiquée sur la figure 3-11.



Le diagramme de cas d'utilisation du nouveau package UC de second rang est donné sur la figure 3-12.

B.A.-BA Stéréotype

Les mots-clés UML comme «include» et «extend» sont notés entre guillemets typographiques. Toutefois, nous pouvons également créer nos propres mots-clés, tels que «fragment» (pour indiquer qu'un cas d'utilisation n'est qu'un fragment factorisé d'autres cas d'utilisation) ou «secondaire» (pour indiquer qu'un cas d'utilisation est moins important que les autres). Ces mots-clés inventés par les utilisateurs s'appellent alors des stéréotypes.

Figure 3-11

Organisation complétée des cas d'utilisation et des acteurs

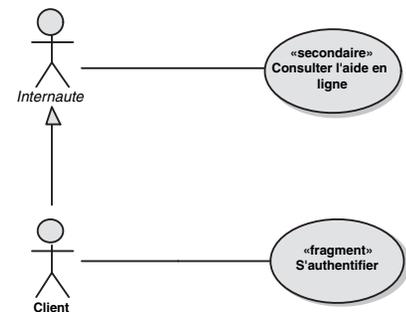


Figure 3-12
Cas d'utilisation de second rang

Classement des cas d'utilisation

Après tout ce travail d'identification des cas d'utilisation, nous pouvons maintenant les classer en tenant compte des deux facteurs suivants :

- 1 la priorité fonctionnelle, déterminée par le service Marketing de jeBouquine ;
- 2 le risque technique, estimé par le chef de projet.

Le tableau suivant illustre cette démarche sur notre étude de cas.

Tableau 3-1 Classement des cas d'utilisation

Cas d'utilisation	Priorité	Risque
Chercher des ouvrages	Haute	Moyen
Gérer son panier	Haute	Bas
Effectuer une commande	Moyenne	Haut
Créer un compte client	Haute	Bas
Consulter ses commandes	Basse	Moyen
Consulter l'aide en ligne	Basse	Bas
Gérer son compte client	Moyenne	Bas
Maintenir le catalogue	Haute	Haut
Maintenir les informations éditoriales	Moyenne	Bas
Maintenir le site	Moyenne	Bas

Nous avons demandé expressément au service Marketing de ne pas classer tous les cas d'utilisation en priorité haute, mais de vraiment faire un effort pour distinguer trois niveaux d'importance. Effectuer une commande est ainsi en priorité moyenne, car l'internaute peut toujours imprimer son devis et commander ensuite par courrier ou fax en envoyant son paiement par la Poste. En revanche, l'accent est mis sur la gestion du catalogue et les capacités de recherche afin de fidéliser l'internaute.

Au niveau des risques techniques, le chef de projet a classé au plus haut la maintenance du catalogue, à cause des problèmes liés à l'intégrité des informations régulièrement mises à jour semi-automatiquement dans la base de données centrale, et à la nécessité absolue d'avoir un catalogue valide et à jour. De même, le passage de commande est noté avec un risque haut à cause des problèmes de confidentialité et de cryptage à résoudre de façon sûre.

Planification du projet en itérations

À partir du classement précédent, le chef de projet a proposé au comité de pilotage le découpage en itérations suivant :

Tableau 3-2 Planification des itérations grâce aux cas d'utilisation

Cas d'utilisation	Priorité	Risque	Itération #
Chercher des ouvrages	Haute	Moyen	2
Gérer son panier	Haute	Bas	4
Effectuer une commande	Moyenne	Haut	3
Créer un compte client	Haute	Bas	5
Consulter ses commandes en cours	Basse	Moyen	7
Consulter l'aide en ligne	Basse	Bas	10
Gérer son compte client	Moyenne	Bas	9
Maintenir le catalogue	Haute	Haut	1
Maintenir les informations éditoriales	Moyenne	Bas	8
Maintenir le site	Moyenne	Bas	6

Un des bons principes du Processus Unifié consiste à identifier et lever les risques majeurs au plus tôt. Le chef de projet doit donc prendre en compte de façon combinée la priorité fonctionnelle et l'estimation du risque :

- Si la priorité est haute et le risque également, il faut planifier le cas d'utilisation dans une des toutes premières itérations (exemple : Maintenir le catalogue).
- Si la priorité est basse et le risque également, on peut reporter le cas d'utilisation à une des toutes dernières itérations (exemple : Consulter l'aide en ligne).
- Les choses se compliquent lorsque les deux critères sont antagonistes ! Le chef de projet doit alors décider en pesant le pour et le contre. Il peut être amené à négocier avec le client pour le convaincre qu'il vaut mieux pour le projet traiter en premier un cas d'utilisation risqué mais peu prioritaire, au lieu d'un cas d'utilisation plus prioritaire mais ne comportant pas de risque.

ATTENTION Processus adaptatif

Ce découpage préliminaire en itérations devra être remis en cause et réévalué à chaque fin d'itération. On qualifie le processus itératif et incrémental d'adaptatif, par opposition à prédictif.

De plus, les itérations ont habituellement une durée fixe, de l'ordre de quatre semaines. Or, il n'est pas certain du tout que chaque cas d'utilisation pourra être conçu, développé et testé dans ce laps de temps. Certains nécessiteront plusieurs itérations, d'autres au contraire pourront être regroupés dans la même.

MÉTHODE Check-list de questions pour le chef de projet

- Quel est le problème métier que vous essayez de résoudre ?
- Quelles fonctionnalités sont essentielles à la solution ?
- Comment la solution proposée peut-elle être décrite de façon à être comprise par des lecteurs techniques et d'autres qui ne le sont pas du tout ?
- Quelles sont les ressources disponibles (temps, personnes, argent) ?
- Comment les exigences doivent-elles être priorisées ?
- Comment peut-on vérifier que le système fonctionnera comme décrit, avec une efficacité et des performances acceptables ?
- Comment peut-on garder la trace des relations de dépendance entre les exigences ?
- Comment peut-on limiter et négocier les demandes de modification de telle sorte que le produit puisse être réalisé dans les temps sans pour autant mécontenter les parties prenantes ?
- Quelle est la procédure de revue et de décision pour les demandes de modification des exigences ?

Traçabilité avec les exigences textuelles

À la fin du chapitre précédent, nous avons saisi un certain nombre d'exigences textuelles dans l'outil RequisitePro (IBM – Rational). Nous avons également commencé à établir des liens de traçabilité entre exigences, par exemples entre exigences de performances et exigences fonctionnelles.

Nous allons maintenant établir la correspondance entre les cas d'utilisation que nous venons d'identifier et les exigences textuelles. Cette étude permet à la fois de valider la complétude des cas d'utilisation, mais aussi d'améliorer celle des exigences textuelles (qui sont parfois contractuelles pour les projets industriels).

Pour réaliser concrètement cette traçabilité, nous allons mettre en œuvre le pont entre les outils RSM (Rational Software Modeler) et RequisitePro. Pour cela, nous allons d'abord ouvrir le référentiel d'exigences à l'intérieur de l'outil de modélisation UML. RSM permet ainsi d'ouvrir une perspective « exigences » en même temps que la perspective « modélisation », ainsi que le montre la figure suivante.

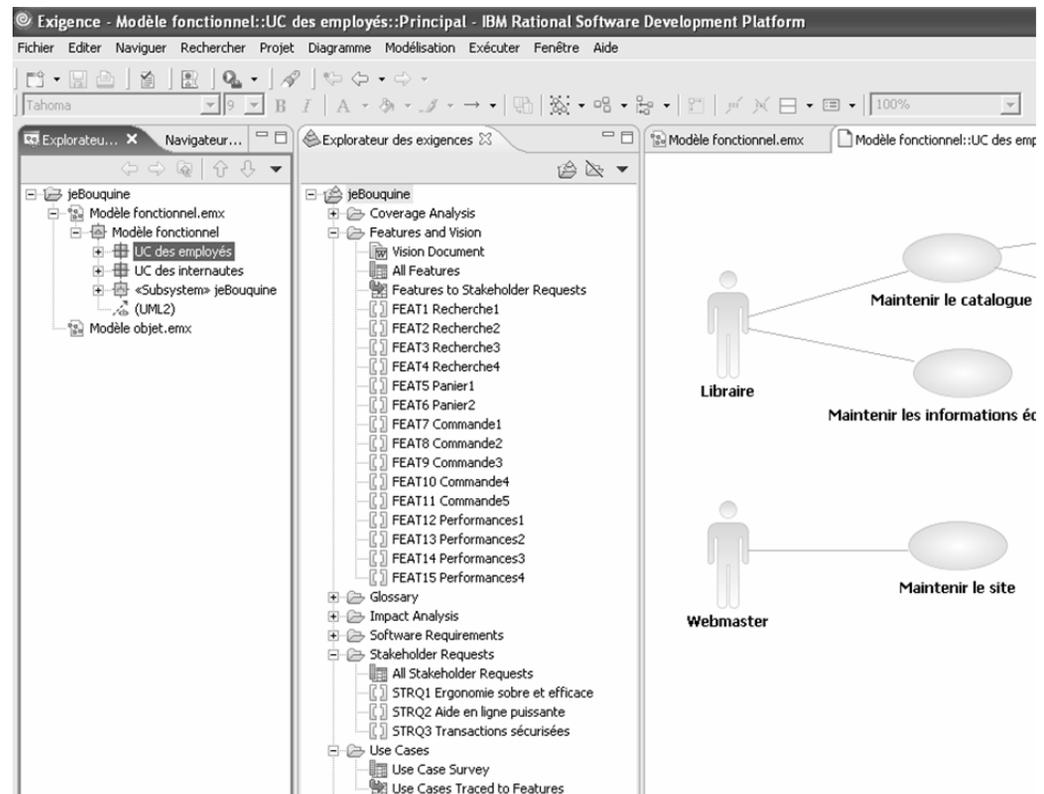


Figure 3–13 Les exigences visibles dans RSM

Nous allons pouvoir ajouter les cas d'utilisation dans RequisitePro par de simples glisser-déplacer de l'explorateur de modèle UML vers l'explorateur d'exigences à l'intérieur de RSM. Le résultat est illustré par la figure 3-14.

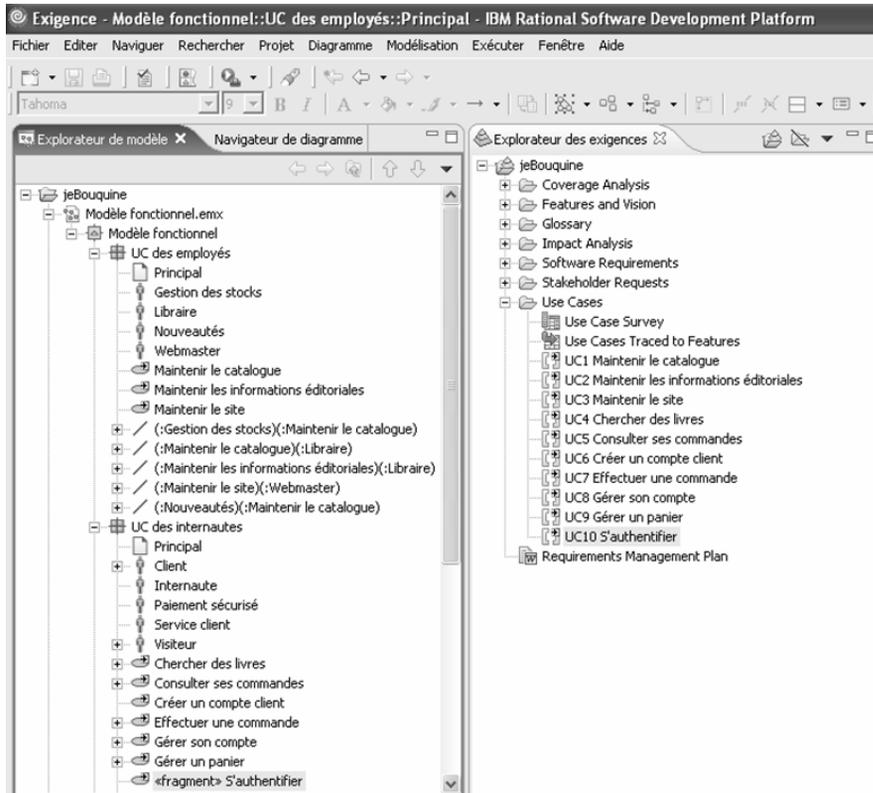


Figure 3-14
Les cas d'utilisation ajoutés
dans l'explorateur d'exigences

Il ne reste plus maintenant qu'à relier les cas d'utilisation aux exigences par des liens de traçabilité dans RequisitePro. Le cas d'utilisation Chercher des livres sera ainsi relié aux exigences appelées RechercheN, ainsi qu'à certaines exigences de performances, etc. RequisitePro permet ensuite de visualiser une matrice de traçabilité telle que celle de la figure 3-15.

Nous pouvons, par exemple, déduire de cette matrice que toutes les exigences textuelles ont bien été tracées par rapport à au moins un cas d'utilisation. Par contre, le cas d'utilisation s'authentifier doit induire de nouvelles exigences, puisqu'il n'est pas relié du tout. Il est tout à fait courant que l'identification des cas d'utilisation amène ainsi à une révision des exigences textuelles.

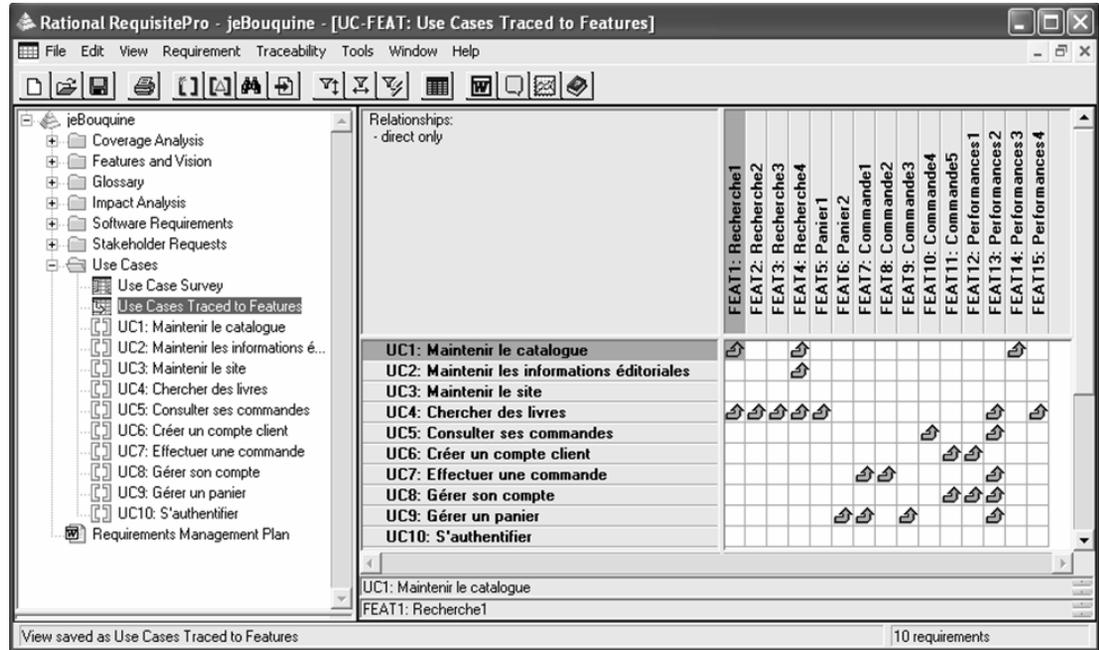


Figure 3–15 Matrice de traçabilité entre cas d'utilisation et exigences