UML en action

De l'analyse des besoins à la conception en Java

Pascal ROQUES
Franck VALLÉE

Deuxième édition 2003

ISBN : 2-212-11213-0

EYROLLES

Chapitre 4

Capture des besoins fonctionnels

Objectifs du chapitre

Ce chapitre traite du rôle que tient UML pour compléter la capture des besoins fonctionnels ébauchée durant l'étude préliminaire. La technique des cas d'utilisation est la pierre angulaire de cette étape. Elle va nous permettre de préciser l'étude du contexte fonctionnel du système, en décrivant les différentes façons qu'auront les acteurs d'utiliser le futur système.

Nous verrons successivement dans ce chapitre comment :

- identifier les cas d'utilisation du système par ses acteurs,
- décrire les cas d'utilisation,
- organiser les cas d'utilisation,
- identifier les classes candidates du modèle d'analyse.

Quand intervient la capture des besoins fonctionnels ?

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire.

Elle est complétée au niveau de la branche droite du Y par la capture des besoins techniques (décrite au chapitre 5) et prépare l'étape suivante de la branche gauche : l'analyse (décrite dans les chapitres 6 à 8).

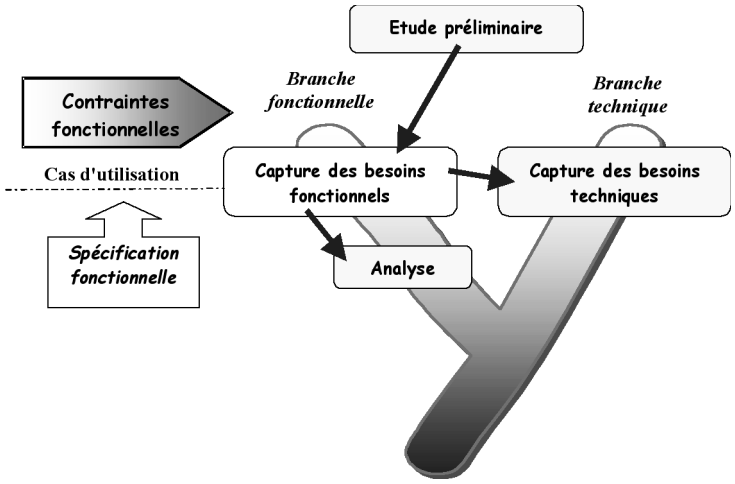


Figure 4-1. : Situation de la capture des besoins fonctionnels dans 2TUP

Éléments mis en jeu

- Messages, acteurs, modèle de contexte dynamique,
- Acteur principal, acteur secondaire,
- Cas d'utilisation, description préliminaire d'un cas d'utilisation,
- Diagramme de cas d'utilisation,
- Fiche de description textuelle d'un cas d'utilisation,
- Scénario, enchaînement, diagramme d'activités,
- Inclusion, extension et généralisation de cas d'utilisation,
- Package de cas d'utilisation,
- Classes candidates, responsabilités, diagramme de classes participantes,
- Traçabilité des cas d'utilisation avec les besoins fonctionnels, itération.

Identifier les cas d'utilisation

Définition

QU'EST-CE QU'UN CAS D'UTILISATION ?

Un *cas d'utilisation* (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant 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.

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. Dans le cadre de la branche fonctionnelle, le cas d'utilisation doit mettre en valeur les interactions métier entre les acteurs et le système. On exprimera donc des actions effectuées dans le cadre du métier de l'utilisateur, par opposition à des manipulations de l'application ou à des comportements techniques. Par exemple, on ne développe ni la manipulation d'une IHM (Interface Homme Machine), ni la gestion d'erreurs matérielles au travers d'un cas d'utilisation.

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 identifié durant l'étude préliminaire, 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.

On se servira avec profit des échanges de *messages* identifiés dans le modèle de contexte dynamique.

Pour chaque cas d'utilisation candidat, il faut :

- vérifier qu'il fournit une valeur ajoutée « notable » à l'acteur, toujours dans le cadre de son métier,
- contrôler qu'un événement externe au système en déclenche l'exécution (sauf exceptionnellement pour des traitements « batch », comme la transmission comptable vers SAP).


Ne pas faire

UN CAS D'UTILISATION N'EST NI UNE TRANSACTION, NI UNE FONCTION !

Une erreur fréquente concernant les cas d'utilisation consiste à vouloir descendre trop bas en termes de granularité, notamment lorsque l'on oublie la valeur métier d'un cas d'utilisation, par opposition aux transactions informatiques produites dans le cadre de la réalisation d'une fonction de l'entreprise. 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'intention fonctionnelle de l'acteur vis-à-vis du système. Le cas d'utilisation ne doit donc pas se réduire systématiquement à une seule séquence, et encore moins à une simple action.

Pour illustrer notre propos, considérez le travail d'un comptable désirant saisir les règlements qu'il a reçus. Son intention est de procéder à la saisie de ces règlements, et il pourra opter entre scanner des chèques, taper manuellement les versements ou saisir une feuille de dépôt de banque. Les trois options donneront lieu à des séquences d'actions différentes du même cas d'utilisation, car la valeur ajoutée est toujours identique dans le métier du comptable. Par ailleurs, le cas d'utilisation comprendra également des déroulements alternatifs ainsi que la gestion de cas d'erreurs, comme nous allons vous le montrer en détail dans ce chapitre.


Conseil

DISTINGUEZ L'ACTEUR PRINCIPAL DES ACTEURS SECONDAIRES !

Nous appelons acteur *principal* celui pour qui le cas d'utilisation produit la plus-value métier. En conséquence, l'acteur principal est la plupart du temps (mais pas forcément, comme dans le cas précité des traitements batch) le déclencheur du cas d'utilisation. Par opposition, nous qualifions d'acteurs secondaires les autres participants du cas d'utilisation. Les acteurs secondaires sont typiquement sollicités à leur tour par le système pour obtenir des informations complémentaires.

Un cas d'utilisation comporte donc :

- un acteur principal (c'est obligatoire),
- d'éventuels acteurs secondaires.


Conseil

NOMMEZ LES CAS D'UTILISATION AVEC UN VERBE À L'INFINITIF SUIVI D'UN COMPLÉMENT !

Une recommandation complémentaire consiste à veiller à bien utiliser le point de vue de l'acteur et non pas celui du système. Par exemple, le cas d'utilisation d'un distributeur de billets par l'acteur « Porteur de CB » doit être intitulé « Retirer de l'argent » (point de vue de l'acteur), et non « Distribuer de l'argent » (point de vue du système).

ÉTUDE DE CAS : LISTE PRÉLIMINAIRE DES CAS D'UTILISATION DE SIVEX

Voici la technique recommandée pour identifier les cas d'utilisation à partir du modèle de contexte : considérez l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message (voir chapitre 3, figure 3-5). En regroupant les intentions fonctionnelles en unités cohérentes, vous obtiendrez les cas d'utilisation recherchés. Le tableau ci-dessous permet d'établir le résultat de ce travail, en montrant le lien entre les cas d'utilisation identifiés, les acteurs principaux et secondaires, et les messages provenant du contexte.

Cas d'utilisation	Acteur principal, acteurs secondaires	Message(s) émis / reçus par les acteurs
Traiter une commande	Réceptionniste	émet : création, modification, annulation commande reçoit : conditions commande
	Client	reçoit : confirmation commande
Gérer les infos clients	Réceptionniste	émet : création client
Consulter les en-cours	Client	reçoit : en-cours
	Réceptionniste	reçoit : en-cours
Gérer la facturation	Comptable	reçoit : factures
Suivre les règlements Comptable		émet : règlement
		reçoit : relance
Planifier une mission	Répartiteur	émet : création, modification mission
	Chauffeur	reçoit : bordereaux mission
Suivre une mission	Chauffeur	émet : arrêt/départ étape, événement mission
	Répartiteur	reçoit : incident mission
	Véhicule	émet : position
Réaliser l'inventaire	Opérateur de quai	émet : début/fin inventaire, pointage colis
Manipuler les colis	Opérateur de quai	émet : pointage colis, identification colis reçoit : listes colis commande, étiquette
Définir le plan de transport	Responsable logistique	émet : création, modification plan reçoit : statistiques transport
Gérer les ressources	Responsable logistique	émet : affectation ressources
Gérer les profils	Administrateur	émet : profil utilisateur

Tableau 4-3 : Liste des acteurs et des messages par cas d'utilisation

Conseil

ÉTABLISSEZ UNE PREMIÈRE DESCRIPTION SUCCINCTE DE CHAQUE CAS D'UTILISATION CANDIDAT !

Chaque cas d'utilisation doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système et les séquences d'actions principales qu'il est susceptible d'effectuer. Ces définitions servent à fixer les idées lors de l'identification des cas d'utilisation et n'ont aucun caractère exhaustif.

EXEMPLES DE DESCRIPTION PRÉLIMINAIRE DE CAS D'UTILISATION

Planifier une mission (Répartiteur) :

- **Intention** : planifier au sein d'une agence une mission nécessaire à la réalisation des commandes en cours ;
- **Actions** : créer une nouvelle mission (regrouper des commandes, affecter des ressources disponibles, établir un parcours, etc.), modifier ou annuler une mission existante.

Suivre une mission (Chauffeur) :

- **Intention** : informer en temps réel de l'état d'avancement de chaque mission en cours ;
- **Actions** : transmettre chaque arrêt/départ d'étape, signaler les événements de mission (acquittement client, panne, retard, absence client, etc.).

Maintenant que nous avons identifié les cas d'utilisation et leurs acteurs, nous allons pouvoir les représenter graphiquement sur un *diagramme de cas d'utilisation*, dont la notation graphique de base est la suivante :

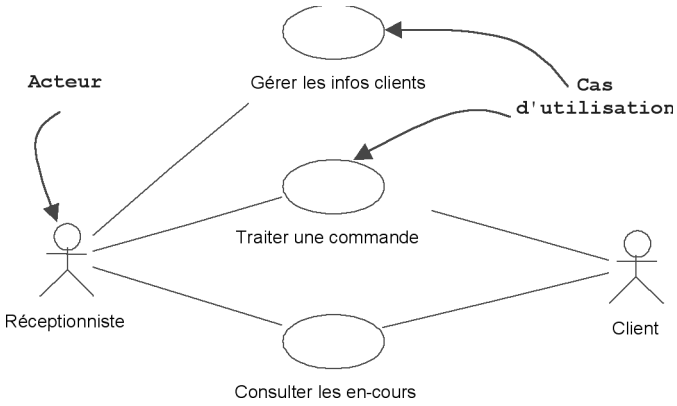


Figure 4-2 : Exemple de diagramme de cas d'utilisation de SIVEx


Conseil

DIAGRAMME DE CAS D'UTILISATION : DÉTAILLEZ LES RÔLES (PRINCIPAL OU SECONDAIRE) ET LE SENS DES ASSOCIATIONS !

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

➢ par défaut, le rôle d'un acteur est « principal » ; si ce n'est pas le cas, indiquez explicitement que le rôle est « secondaire » sur l'association, du côté de l'acteur ;

➢ si un acteur a pour rôle unique de consommer des informations du système, sans modifier l'état de celui-ci au niveau métier, représentez cette particularité en ajoutant une flèche vers l'acteur sur son association avec le cas d'utilisation ;

➢ si un acteur a pour rôle unique de fournir des informations au système sans en recevoir, représentez cette particularité en ajoutant sur l'association une flèche vers le cas d'utilisation.

Dans l'exemple précédent, nous avons trois cas de figure différents :

- *Gérer les infos clients*, qui n'a qu'un seul acteur principal : le réceptionniste ;
- *Traiter une commande*, qui a deux acteurs : le réceptionniste qui est principal, et le client qui est secondaire ;
- *Consulter les en-cours*, qui a également deux acteurs. Tous deux peuvent accéder à la même fonctionnalité métier. Dans ce cas, l'acteur principal est celui qui tire réellement bénéfice des résultats du cas d'utilisation. Il s'agit en fait du client qui passe soit directement par Internet, soit par l'intermédiaire du réceptionniste pour obtenir l'information.

UML ne comporte pas de notation standard pour distinguer graphiquement ces trois cas. Mais, avec les conventions que nous avons répertoriées plus haut, le diagramme devient :

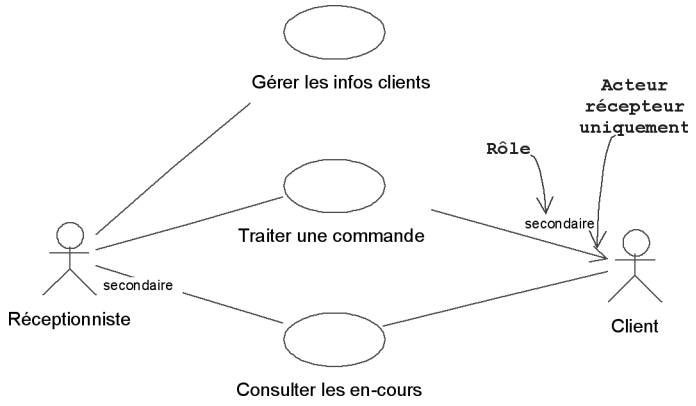


Figure 4-3. : Ajouts graphiques sur le diagramme de cas d'utilisation de SIVEx

Ne pas faire

NE RÉINVENTEZ PAS LA DÉCOMPOSITION FONCTIONNELLE !

Paradoxalement, malgré l'apparente simplicité du concept de cas d'utilisation, et son acceptation immédiate dans la plupart des méthodes objet, il existe des risques importants de mauvais emploi des cas d'utilisation !

Ces risques sont principalement liés à :

- leur nature fonctionnelle, et non objet,
- la difficulté de savoir à quel niveau de détail s'arrêter.

Un nombre trop important de cas d'utilisation est en général le symptôme d'une décomposition fonctionnelle descendante hiérarchique. Nous avons vu plusieurs projets s'engluer ainsi dans des tentatives de structuration des cas d'utilisation en « sous-cas d'utilisation » avec plusieurs niveaux de décomposition, pour tenter d'arriver à des sortes de transactions élémentaires.

La décomposition fonctionnelle, très en vogue dans les années 80, a montré ses limites dans la pratique, en particulier sur les gros projets. Il ne s'agit donc pas de réintroduire, avec la décomposition des cas d'utilisation, les problèmes connus que l'approche orientée objet permet justement d'éviter !

Fondamentalement, un cas d'utilisation sera réalisé par une collaboration d'objets du système, mais cette correspondance n'est absolument pas bijective. En effet, un objet est souvent impliqué dans la réalisation de plusieurs cas d'utilisation. Ainsi, sur un gros projet, si l'on se sert uniquement des cas d'utilisation pour découper le travail entre les équipes de développement, on aboutit inévitablement à ce que ces équipes construisent en parallèle les mêmes classes, en introduisant des incohérences. Un regroupement de classes suivant leur implication par cas d'utilisation a donc peu de chance de perdurer dans l'organisation des classes d'analyse. Ces dernières s'organiseront dans le modèle structurel d'analyse qui vise notamment la modularité des concepts manipulés dans le système. C'est cette capacité à organiser, partager, isoler et réutiliser des concepts qui manquait cruellement à l'approche par décomposition fonctionnelle.

N'oubliez pas cependant que les cas d'utilisation ne constituent pas une fin en soi. Leur objectif est de :

- dialoguer avec le client,
- analyser les besoins métier,
- disposer d'un support d'analyse de la valeur,
- aider à démarrer l'analyse orientée objet en identifiant les classes candidates.

LIMITEZ À 20 LE NOMBRE DE VOS CAS D'UTILISATION !

Le niveau de granularité des cas d'utilisation étant comme on l'a vu très variable, cette limite arbitraire oblige à ne pas se poser trop de questions philosophiques et à rester synthétique. Dans les paragraphes suivants, nous expliquons qu'un cas d'utilisation décrit en réalité un ensemble de scénarios. Il est ainsi courant d'avoir une quinzaine de cas d'utilisation, comprenant chacun une dizaine de scénarios. Vous pouvez considérer ces ordres de grandeur comme une limite pratique qui vous aidera à mieux situer la frontière entre cas d'utilisation et scénario.

Nous conseillons donc d'identifier un nombre restreint de « grands » cas d'utilisation, alors que certains auteurs [Texel 97] préconisent au contraire de nombreux « petits » cas d'utilisation, mais avec les risques identifiés au paragraphe précédent.

Décrire les cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'interactions entre le système et ses acteurs. Pour décrire la dynamique du cas d'utilisation, le plus naturel consiste à recenser toutes les interactions de façon textuelle. Le cas d'utilisation doit par ailleurs avoir un début et une fin clairement identifiés. Il doit préciser quand ont lieu les interactions entre acteurs et système, et quels sont les messages échangés. Il faut également préciser les variantes possibles, telles que les différents cas nominaux, les cas alternatifs, les cas d'erreurs, tout en essayant d'ordonner séquentiellement les descriptions, afin d'améliorer leur lisibilité. Chaque unité de description de séquences d'actions est appelée *enchaînement.* Un *scénario* représente une succession particulière d'enchaînements, qui s'exécute du début à la fin du cas d'utilisation.

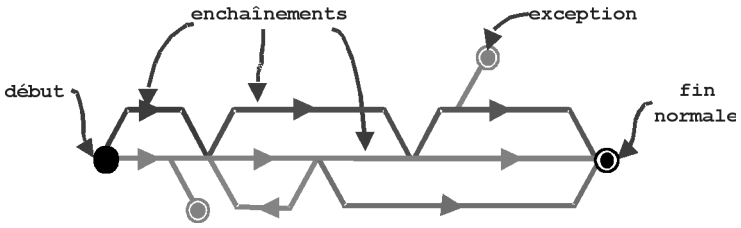


Figure 4-4. : Représentation des variantes d'un cas d'utilisation

Les scénarios sont aux cas d'utilisation ce que les objets sont aux classes : on peut considérer qu'un scénario est une instance particulière d'un cas d'utilisa-

tion. L'acteur principal d'un cas d'utilisation dispose donc de l'ensemble des enchaînements pour réaliser une certaine tâche métier. Les exceptions décrivent les interruptions possibles d'exécution empêchant l'acteur d'obtenir sa plus-value métier.


Conseil

CAS D'UTILISATION : UTILISEZ LE STYLE DE DESCRIPTION ADAPTÉ !

N'oubliez pas qu'un cas d'utilisation a pour fonction de décrire une utilisation du système par un acteur particulier. La façon dont vous allez procéder à cette description dépend de la raison qui vous conduit à l'effectuer. Vous la présenterez d'une certaine manière à votre client, parce que vous espérez la lui faire valider. Vous l'exposerez d'une autre manière à votre équipe d'analystes et de concepteurs, parce que vous essayez de leur donner suffisamment d'informations pour qu'ils puissent passer aux phases suivantes.


Ne pas faire

CAS D'UTILISATION : NE MÉLANGEZ PAS L'IHM ET LE FONCTIONNEL !

Une erreur fréquente concernant les cas d'utilisation consiste à les rendre dépendants d'un choix prématuré d'interface homme-machine. Il faudra entièrement les redocumenter à chaque évolution d'interface, alors qu'il s'agit en fait toujours du même cas d'utilisation fonctionnel. Encore une fois, l'usage des cas d'utilisation, dans le cadre de la branche fonctionnelle du processus 2TUP vise exclusivement la description du métier des acteurs.

Pour illustrer notre recommandation en faveur du développement des cas d'utilisation métier, nous préférons exprimer un enchaînement de la façon suivante :

- « Lors d'une première prise de commande, le réceptionniste doit enregistrer les caractéristiques du nouveau client dans le système » ;

à l'inverse de :

- « Le réceptionniste doit saisir le nom du client sur 8 caractères maximum, appuyer sur ENTER, puis saisir le prénom sur 8 caractères maximum et appuyer sur ENTER » ;

ou bien de :

- « Le réceptionniste enregistre au moyen du dispositif de reconnaissance vocale le nom du client, son prénom, son adresse, son code postal ».

La règle consiste à ne pas alourdir la description des séquences logiques d'interactions entre les acteurs et le système de considérations techniques d'IHM, et de manière plus générale de considérations qui concernent la mise en application du métier sous forme informatique. Cela n'empêche pas d'annexer à la description du cas d'utilisation une proposition d'IHM si le client le souhaite, ou mieux, un descriptif des besoins d'IHM. Cette dernière proposition permet de ne pas figer

trop rapidement l'IHM, ce qui risquerait de multiplier les styles et de nuire à l'ergonomie d'ensemble. On développera en revanche une consolidation de l'application, en fonction des besoins d'IHM, en procédant à une analyse de la valeur sur les besoins réels du métier et en faisant intervenir un spécialiste des IHM pour développer une maquette globale des écrans.


Conseil

CAS D'UTILISATION : COMMENT STRUCTURER LES FICHES DE 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),**

- inclut titre, but, résumé, dates, version, responsable, acteurs...

➢ **Description des enchaînements (obligatoire),**

- décrit les enchaînements nominaux, les enchaînements alternatifs, les enchaînements d'exception, mais aussi les préconditions, et les postconditions.

➢ **Besoins d'IHM (optionnel),**

- ajoute éventuellement les contraintes d'interface homme-machine : ce qu'il est nécessaire de montrer, en conjonction avec les opérations que l'utilisateur peut déclencher…

➢ **Contraintes non-fonctionnelles (optionnel).**

- ajoute éventuellement les informations suivantes : fréquence, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, performances, concurrence, etc. Ces informations peuvent servir à mieux évaluer les contraintes techniques, et pourront améliorer, par consolidation, la capture des besoins opérée en parallèle par l'architecte technique (voir chapitre 5).

ÉTUDE DE CAS : CAS D'UTILISATION « PLANIFIER UNE MISSION »

Sommaire d'identification :

Titre : Planifier une mission

But : planifier une mission d'une agence à partir de la connaissance du plan de transport, des ressources disponibles et des commandes à assurer quotidiennement.

Résumé : création d'une nouvelle mission d'enlèvement, de livraison ou de traction à partir des commandes confirmées. Modification ou annulation de mission.

Acteurs : Répartiteur (principal), *Chauffeur (secondaire).*

Date de création : 02/02/02 **Date de mise à jour** : 24/07/02

Version : 2.1 **Responsable** : Pascal Roques

Description des enchaînements :

Préconditions :

1. Le répartiteur est authentifié.

2. Il existe au moins une commande confirmée à planifier.

3. Au moins un chauffeur et un véhicule sont disponibles.

4. Les parcours prédéfinis sont disponibles (plan de transport).

Enchaînements nominaux :

Ce cas d'utilisation commence lorsque le répartiteur demande au système de créer une nouvelle mission.

Enchaînement (a) Créer une mission en construction

Le répartiteur fournit un nom d'identification et établit obligatoirement la nature (enlèvement, livraison ou traction) de la mission qu'il veut créer.

S'il s'agit d'une mission de traction, le répartiteur doit indiquer une agence principale de destination.

Enchaînement (b) Affecter les commandes

Le répartiteur affecte les commandes à une mission. Le système évalue au fur et à mesure des affectations le tonnage et la durée estimés de la mission.

Enchaînement (c) Affecter les ressources

Le répartiteur affecte un véhicule et un chauffeur à la mission, en fonction du tonnage évalué.

Si la mission dépasse la capacité du véhicule alors il faut exécuter
[Exception 1 : dépassementTonnage].

Si le chauffeur n'a pas les qualifications requises pour conduire le véhicule alors il faut exécuter
[Exception 2 : chauffeurNonQualifié].

Si le tonnage de réserve de l'agence est entamé alors il faut exécuter
[Exception 3 : tonnageReserveEntamé].

Enchaînement (d) Définir le trajet

Le répartiteur propose l'ordre des étapes à suivre. Pour une traction, il suffit de choisir un parcours parmi ceux prévus dans le plan de transport. Pour une livraison ou un enlèvement, le répartiteur peut avoir à choisir plusieurs parcours pour rejoindre tous les sites étapes de la mission.

Enchaînement (e) Valider une mission en construction

Le répartiteur valide une mission en construction : il doit alors préciser l'heure de départ prévue. Le système édite alors les bordereaux de mission. Ces bordereaux contiennent une fiche de description et de réception par commande ainsi qu'une feuille de route décrivant les étapes et les horaires estimés.

Enchaînement alternatifs :

Enchaînement (f) Modifier une mission en construction

Le répartiteur désaffecte une commande, ou affecte à nouveau le véhicule et le chauffeur d'une mission en construction. Le répartiteur modifie également à son gré l'ordre des étapes proposé pour la mission.

Enchaînement (g) Modifier une mission validée

Le répartiteur peut encore modifier une mission au minimum 1 heure avant son départ. Toute modification d'une mission validée entraînant son invalidation, il doit donc ensuite la valider à nouveau en précisant une heure de départ.

Enchaînement (h) Annuler une mission

Le répartiteur annule une mission non encore validée ou une mission validée au minimum 1 heure avant son départ.

Ce cas d'utilisation se termine lorsque le répartiteur a :

• amené la mission jusqu'à son départ,

• ou bien annulé la mission.Exceptions :

Exception :

[Exception 1 : dépassementTonnage] ou **[Exception 2 : chauffeurNonQualifié]** : la mission est marquée en anomalie tant que le répartiteur n'a pas corrigé l'erreur. Il ne peut plus valider une telle mission

[Exception 3 : tonnageReserveEntamé] : un message d'erreur reste affiché sur l'écran du répartiteur, tant que le tonnage de réserve n'est plus assuré.

Postconditions :

1. Le véhicule affecté à une mission validée possède la capacité de tonnage nécessaire.

2. Le chauffeur affecté à une mission validée possède la qualification nécessaire.

3. Les commandes d'une mission validée sont considérées comme programmées du point de vue du réceptionniste.

Besoins d'IHM :

❍ Pour lister les commandes concernant l'agence

Afin d'établir la planification de ses missions, le répartiteur doit pouvoir répertorier les commandes que son agence doit assurer.

Il doit pouvoir filtrer ou ordonner cette liste suivant :

- le type de commande (enlèvement, traction ou livraison),
- le poids,
- le site desservi,
- l'affectation à une mission ou non,
- la tarification urgent/non urgent.

Chaque ligne de la liste représente une commande et regroupe l'ensemble des informations de filtre.

Une couleur différente doit permettre de distinguer les commandes affectées de celles qui ne le sont pas.

Le répartiteur peut affecter les commandes aux missions depuis cette liste.

❍ Pour lister les véhicules et les chauffeurs disponibles à l'agence

Le répartiteur doit pouvoir répertorier les véhicules et les chauffeurs dont il dispose dans l'agence. Il peut filtrer ou ordonner ces deux listes par tonnage pour les véhicules, par qualification pour les chauffeurs.

Le répartiteur peut affecter les ressources aux missions depuis ces deux listes.

❍ Pour lister les missions définies dans l'agence

Le répartiteur doit pouvoir répertorier les missions déjà définies pour l'agence. Il peut filtrer ou ordonner cette liste suivant :

- l'état de validation,
- la nature (enlèvement, livraison, traction),
- les sites desservis,
- le tonnage déjà affecté.

Une couleur différente doit permettre de distinguer les missions validées de celles qui ne le sont pas.

❍ Pour consulter ou modifier une mission

Le répartiteur dispose d'une fiche par mission. Plusieurs fiches peuvent être ouvertes simultanément. Cette fiche récapitule :

- la nature de la mission,
- le tonnage déjà affecté,
- l'heure de départ si la mission est validée,
- les ressources affectées à la mission,
- la liste ordonnée des étapes avec les commandes concernées et l'horaire de passage estimé.

Depuis cette fiche, le répartiteur peut modifier les ressources et les commandes affectées, permuter l'ordre des étapes, saisir les estimations de parcours manquantes, et valider la mission.

Contraintes non fonctionnelles :

Contrainte	Descriptif
Temps de réponse	L'interface du répartiteur doit réagir en l'espace de deux secondes au maximum.
Concurrence	Les validations de mission doivent être notifiées par un message d'avertissement aux autres lecteurs potentiels de la mission.
Fréquence	Non applicable.
Volumétrie	Une mission représente en moyenne 0,5 Ko de données. Le nombre moyen estimé de missions par mois est de 46 000, et leur durée de rétention doit être de 6 mois.
Disponibilité	Le système est accessible aux répartiteurs 6 jours sur 7, aux heures d'ouverture des agences.

Intégrité	Non applicable dans la mesure où les missions d'une agence ne sont accessibles qu'au seul répartiteur en modification.
Confidentialité	Les répartiteurs sont identifiés par le système en fonction de leur nom, de leur mot de passe et du rôle qu'ils détiennent dans l'agence.
Intégration des applications	Lors de la validation d'une mission, toutes les commandes affectées passent automatiquement à l'état « shipped » dans le CRM SIEBEL. Le numéro de mission est transmis comme référence du shipping.

COMPLÉTEZ LES DESCRIPTIONS TEXTUELLES AVEC DES DIAGRAMMES DYNAMIQUES SIMPLES !

Pour documenter les cas d'utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement et précisément avec les utilisateurs. La description textuelle est également l'occasion de s'entendre sur la terminologie employée, ainsi que d'identifier le contexte d'exécution de l'un ou de l'autre des enchaînements. En revanche, le texte présente des désavantages puisqu'il est difficile de montrer comment les enchaînements se succèdent ; en outre la maintenance des évolutions s'avère souvent périlleuse.

Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques, qui apporteront un niveau supérieur de formalisation. À vous de décider en fonction de votre contexte si vous montrez ces diagrammes au futur utilisateur, ou si vous les utilisez uniquement comme support d'analyse pour lui poser des questions supplémentaires, et ainsi mieux valider votre texte.

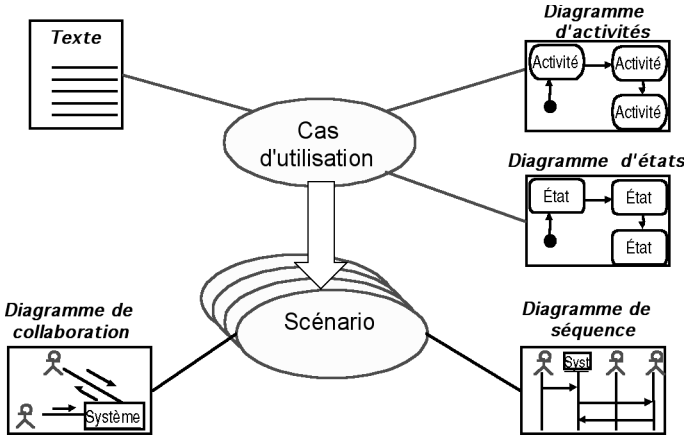


Figure 4-5. : Types de diagrammes dynamiques utilisables pour documenter les cas d'utilisation

Les critères de choix entre les différents types de diagrammes dynamiques utilisables sont énumérés de façon plus détaillée ci-après :

Pour documenter les cas d'utilisation :

- *Le diagramme d'activités* est celui que nous recommandons particulièrement, car il permet de consolider les enchaînements de la fiche textuelle, comme nous l'avions représenté informellement sur la figure 4-4 . Ce diagramme est également très utile en cas d'activités parallèles. De plus, les utilisateurs le comprennent aisément, car il ressemble à un organigramme traditionnel. Il permet enfin d'identifier d'un seul coup d'œil la famille des scénarios d'un cas d'utilisation qui décrivent toutes les réactions du système. Il suffit en effet de dessiner les différents chemins du diagramme d'activités qui passent par toutes les transitions entre activités.
- Le *diagramme d'états* se prête mieux à la modélisation d'un déroulement événementiel. Il est plus complexe à comprendre pour les utilisateurs du monde de la gestion.

Pour illustrer des scénarios particuliers :

- le *diagramme de séquence* est une bonne illustration. Il est facilement compris par les utilisateurs. De plus, il servira de base aux diagrammes de séquence dont nous parlerons au chapitre 8, et qui mettront en jeu des objets du système.
- le *diagramme de collaboration* est une autre illustration possible. Il est cependant ici moins utile que le précédent pour les utilisateurs, car il rend la séquence moins claire, sans apporter de véritable plus-value au diagramme de séquence.

ÉTUDE DE CAS : CAS D'UTILISATION « PLANIFIER UNE MISSION »

Au cas d'utilisation décrit précédemment, nous allons ajouter un diagramme d'activités qui va préciser l'enchaînement des actions à entreprendre, avec les branchements conditionnels et les boucles possibles. Le cas d'utilisation est exprimé du point de vue de l'utilisateur ; par conséquent les activités correspondent à celles effectuées par l'utilisateur avec le système. Il faut donc interpréter l'activité « estimer parcours » non comme une activité interne du système, mais comme une demande directement déclenchée par l'utilisateur et suivie de son attente de la réponse.

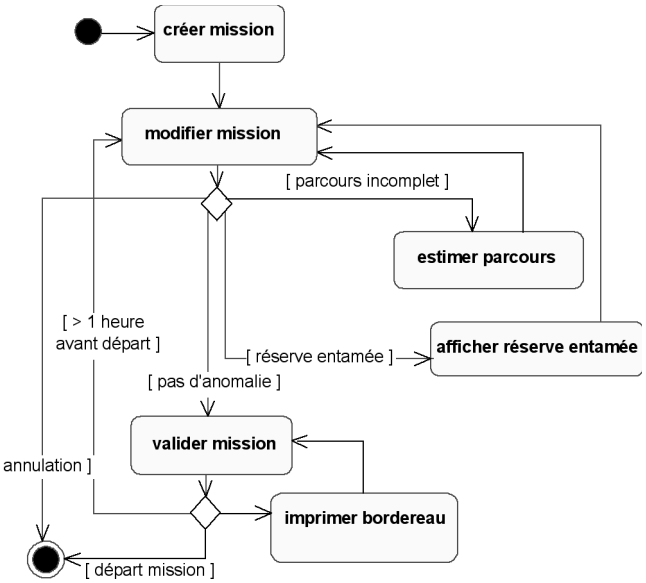


Figure 4-6. : Diagramme d'activités du cas « Planifier une mission »

Nous pouvons également ajouter un diagramme de séquence montrant le scénario nominal de création d'une nouvelle mission. Notez le positionnement de l'acteur principal à gauche du système (vu comme une boîte noire) et de l'acteur secondaire à droite.

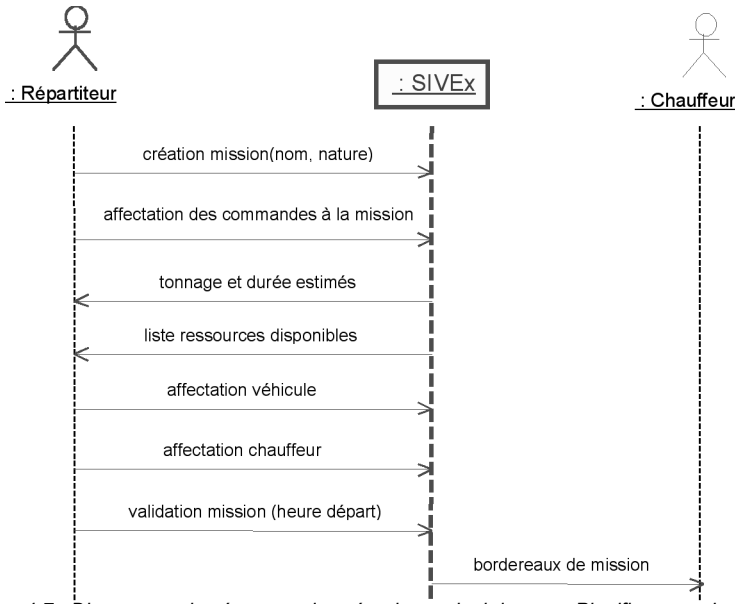


Figure 4-7 : Diagramme de séquence du scénario nominal du cas « Planifier une mission »

Organiser les cas d'utilisation

On peut organiser les cas d'utilisation de deux façons différentes et complémentaires :

- en ajoutant des relations d'inclusion, d'extension, et de généralisation entre les cas d'utilisation ;
- en les regroupant en packages, afin de définir des blocs fonctionnels de plus haut niveau.

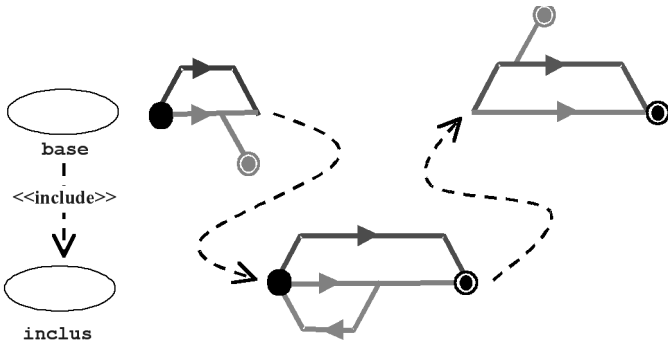
LES RELATIONS POSSIBLES ENTRE CAS D'UTILISATION

UML 1.4 définit trois types de relations standardisées entre cas d'utilisation, détaillées ci-après :

- une relation d'inclusion, formalisée par un mot-clé <<include>>,
- une relation d'extension, formalisée par un mot-clé <<extend>>,
- une relation de généralisation/spécialisation.

LA RELATION «INCLUDE» ENTRE CAS D'UTILISATION

Relation d'inclusion : le cas de base en incorpore explicitement un autre, à un endroit spécifié dans ses enchaînements. Le cas d'utilisation inclus n'est jamais exécuté seul, mais seulement en tant que partie d'un cas de base plus vaste.



Remarquez que dans une relation « include », le cas d'utilisation de base utilise systématiquement les enchaînements provenant du cas inclus. On utilise fréquemment cette relation pour éviter de décrire plusieurs fois le même enchaînement, en factorisant le comportement commun dans un cas d'utilisation à part.

Ne pas faire

INCLUSION : PAS DE DÉCOMPOSITION FONCTIONNELLE !

Une habitude malheureusement assez répandue consiste à utiliser la relation d'inclusion pour décomposer un cas d'utilisation en « sous-cas d'utilisation », retombant dans le travers de la décomposition fonctionnelle évoqué précédemment.

ÉTUDE DE CAS : INCLUSION DU CAS D'UTILISATION « S'AUTHENTIFIER »

Si l'on examine en détail les descriptions textuelles des cas d'utilisation du système SIVEx, on s'aperçoit rapidement que dans toutes les préconditions, on a spécifié que l'acteur principal du cas d'utilisation doit s'être authentifié.

En fait, le processus d'authentification implique un flot d'événements entre l'acteur et le système : saisie d'un login, puis d'un mot de passe, avec les différents cas d'erreur possibles. Cet enchaînement fait partie de la capture des besoins, puisqu'il est visible de l'utilisateur final ; néanmoins il n'est pas de même niveau que les cas d'utilisation que nous avons déjà identifiés. Il correspond tout à fait à la notion de cas d'utilisation inclus, ne s'exécutant jamais seul, mais seulement lorsqu'il est appelé par un cas d'utilisation plus large.

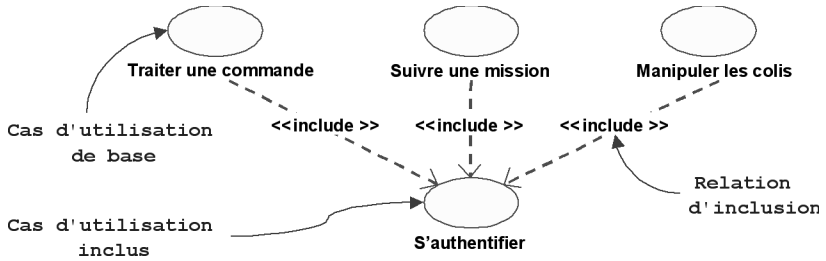
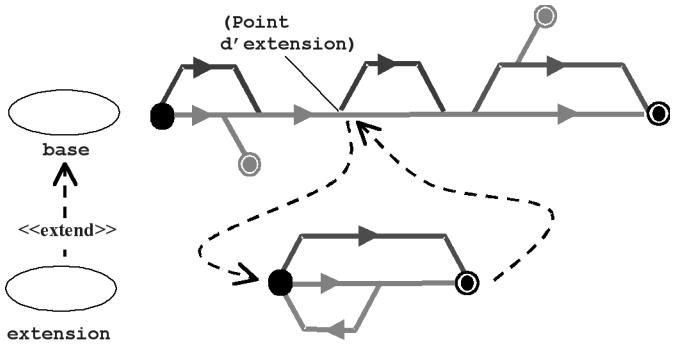


Figure 4-9. : Relation <<include>> entre cas d'utilisation

Définition

LA RELATION « EXTEND » ENTRE CAS D'UTILISATION

Relation d'extension : le cas de base en incorpore implicitement un autre, à un endroit spécifié indirectement dans celui qui étend. Le cas de base peut fonctionner tout seul, mais il peut également être complété par un autre, sous certaines conditions, et uniquement à certains points particuliers de son flot d'événements appelés points d'extension.



Notez que dans une relation « extend », le cas d'utilisation de base recourt optionnellement aux enchaînements provenant du cas d'extension. On utilise principalement cette relation pour séparer le comportement *optionnel* (les variantes) du comportement obligatoire.

ÉTUDE DE CAS : EXTENSION DU CAS « TRAITER UNE COMMANDE »

Dans le cas d'utilisation « Traiter une commande », un des enchaînements principaux consiste à créer une nouvelle commande. Or, si le client est inconnu du système SIVEx, le réceptionniste va devoir interrompre son processus de création de commande pour tenter auparavant de créer un nouveau client. Si ce processus se déroule sans encombre, il pourra alors continuer sa création de commande. Le processus de création de client, pour sa part, fait partie intégrante du cas d'utilisation « Gérer les infos clients ». Il est donc intéressant de préciser cette relation d'extension entre les deux cas d'utilisation.

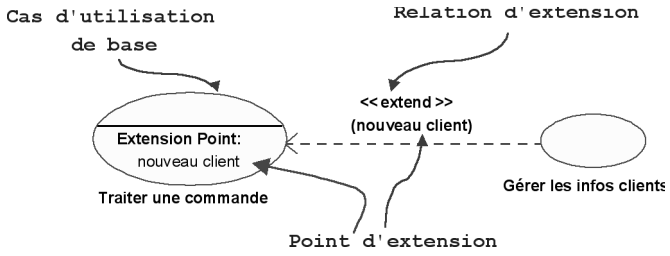


Figure 4-11. : Relation <<extend>> entre cas d'utilisation

LA RELATION DE GÉNÉRALISATION ENTRE CAS D'UTILISATION

Les cas d'utilisation peuvent être hiérarchisés par généralisation/ spécialisation. Les cas d'utilisation descendants héritent de la sémantique de leur parent. Ils peuvent comprendre des interactions spécifiques supplémentaires, ou modifier les interactions héritées.

ÉTUDE DE CAS : SPÉCIALISATION DU CAS « PLANIFIER UNE MISSION »

Si nous reprenons la description textuelle du cas d'utilisation « Planifier une mission », nous pouvons sans conteste identifier de légères variations dans les enchaînements, suivant le type de la mission traitée (enlèvement, traction, livraison).

Par exemple, dans l'enchaînement (a), un comportement supplémentaire pour les missions de traction est précisé : « S'il s'agit d'une mission de traction, le répartiteur doit indiquer une agence principale de destination ». De même, l'enchaînement (e) conduit à l'édition de bordereaux de mission différents en fonction de la nature de la mission.

Nous pouvons donc éventuellement identifier des cas d'utilisation spécialisés suivant le type de mission, avec, en l'occurrence, un cas d'utilisation général abstrait (il ne s'instancie pas directement, mais uniquement par le biais de l'un de ses cas spécialisés).

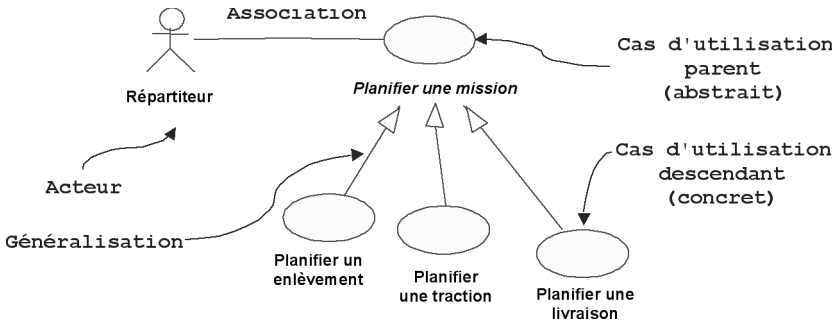


Figure 4-12 : Relation de généralisation entre cas d'utilisation

Conseil

VOUS POUVEZ AUSSI GÉNÉRALISER LES ACTEURS !

Si un ensemble d'acteurs communiquent de la même façon avec certains cas d'utilisations, on peut créer un acteur généralisé (souvent *abstrait*), qui permettra de factoriser ce rôle commun. Les acteurs spécialisés héritent alors des associations de l'acteur ancêtre.

ÉTUDE DE CAS : ACTEUR GÉNÉRALISÉ « UTILISATEUR »

Revenons sur le processus d'authentification identifié plus haut. Il est inclus dans tous les autres, et implique le même type de flot d'événements entre chaque acteur et le système. Pour le représenter sur un diagramme de cas d'utilisation, le plus efficace consiste à créer un acteur abstrait « Utilisateur », associé au cas d'utilisation « S'authentifier », qui généralise tous les acteurs de SIVEx, sauf bien entendu le véhicule.

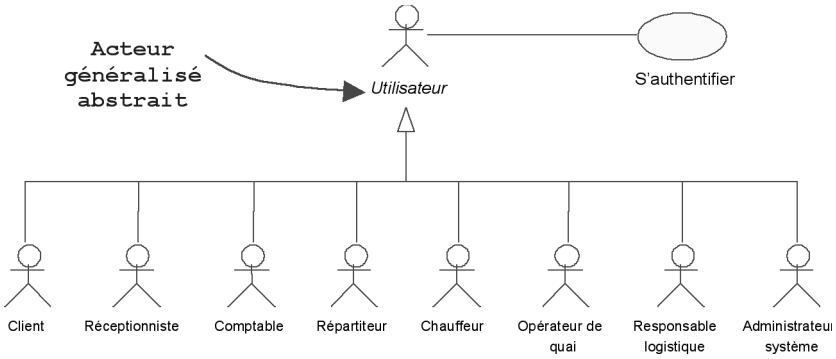


Figure 4-13. : Acteur généralisé « Utilisateur »

Pour définir la stratégie de regroupement des cas d'utilisation pour un projet, il convient de recourir à la liste suivante de critères :

- par domaine d'expertise métier : le plus intuitif et souvent le plus efficace. Il facilite la spécialisation des analystes et permet d'organiser la disponibilité des différents experts ;
- par acteur : simple à mettre en œuvre uniquement si chaque cas d'utilisation est relié à un et un seul acteur, sinon il s'apparente souvent au critère précédent ;
- par lot de livraison : dans le cadre d'un développement itératif et incrémental, il est intéressant de regrouper dans un même package les cas d'utilisation qui seront livrés ensemble au client. Du coup, la structuration peut être très différente de celle obtenue en appliquant le premier critère.

Le mécanisme générique de regroupement d'éléments en UML s'appelle le *package*. Nous allons y recourir dans cette activité, afin de structurer notre ensemble de cas d'utilisation.

Définition

QU'EST-CE QU'UN PACKAGE ?

Un package UML représente un espace de nommage qui peut contenir :

- des éléments d'un modèle,
- des diagrammes qui représentent les éléments du modèle,
- d'autres packages.

Les éléments contenus dans un package :

- doivent représenter un ensemble fortement cohérent**,**
- sont généralement de même nature et de même niveau sémantique.

Nous allons appliquer ces principes aux cas d'utilisation définis pour SIVEx.

ÉTUDE DE CAS : STRUCTURATION DES CAS D'UTILISATION

Le critère de regroupement retenu pour le système SIVEx est le premier cité, soit le domaine d'expertise métier. Il correspond également à un découpage par ensemble d'acteurs fortement reliés. Si nous reprenons le tableau préliminaire, en affectant chaque cas d'utilisation à un package, nous obtenons ce qui suit :

Cas d'utilisation	Acteurs	Package
Traiter une commande	Réceptionniste	Gestion clientèle
	Client	
Gérer les infos clients	Réceptionniste	
Consulter les en-cours	Client	
	Réceptionniste	
Planifier une mission	Répartiteur	Gestion missions
	Chauffeur	
Suivre une mission	Chauffeur	
	Répartiteur	
	Véhicule	
Gérer la facturation	Comptable	Comptabilité
Suivre les règlements	Comptable	
Réaliser l'inventaire	Opérateur de quai	Traitement colis
Manipuler les colis	Opérateur de quai	
Définir le plan de transport	Responsable logistique	Logistique
Gérer les ressources	Responsable logistique	
Gérer les profils	Administrateur	Services support

Tableau 4-4 : Liste des cas d'utilisation et de leurs acteurs par package

Chaque package de cas d'utilisation occasionne la création d'un diagramme. Voici les deux plus intéressants :

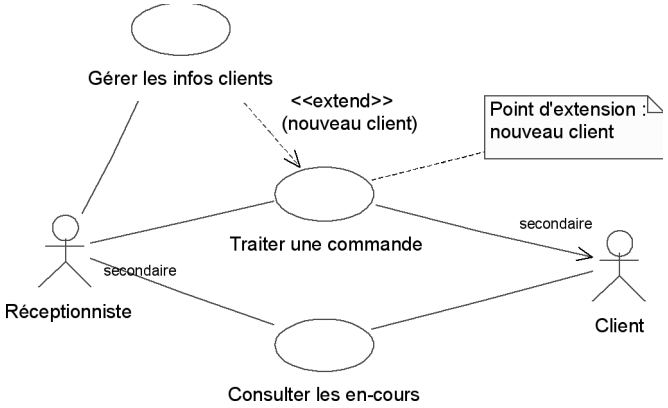


Figure 4-14. : Diagramme de cas d'utilisation du package « Gestion clientèle »

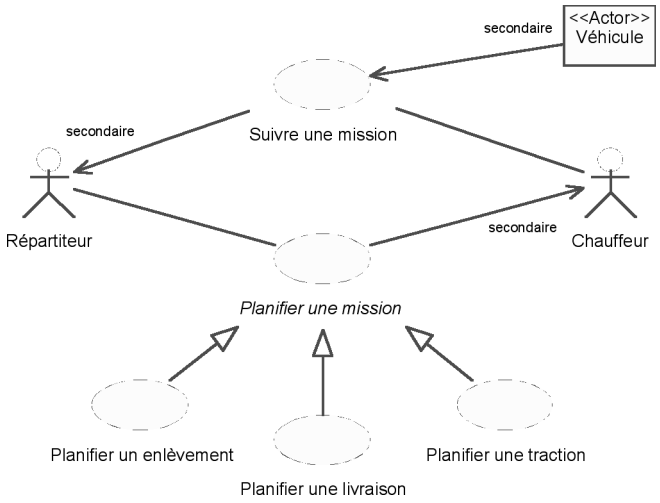


Figure 4-15. : Diagramme de cas d'utilisation du package « Gestion missions »

Décrire les cas d'utilisation en identifiant les flux entre applications

La description des cas d'utilisation doit être l'occasion d'identifier les flux de type EAI qui servent à la synchronisation entre les différentes applications participant au système. Une première technique consiste à ajouter une contrainte non fonctionnelle nommée « intégration des applications » pour décrire les échanges entre applications lors du déroulement du cas d'utilisation ; vous avez pu en avoir un aperçu précédemment, lors de la description du cas d'utilisation « Planifier une mission ».

Cependant, l'identification des flux de synchronisation est souvent le sujet d'une analyse attentionnée dans la mesure où l'on aborde ici des aspects d'urbanisme qui, au cœur du système d'information, touchent un domaine que doivent absolument valider les experts métier. En conséquence, la description des processus métier implique de montrer la répartition des activités entre les applications. En fonction du type de projet, cette description peut être utilisée soit au moment de la modélisation métier, soit comme nous allons l'illustrer, sur le diagramme d'activités d'un cas d'utilisation.

ÉTUDE DE CAS : CAS D'UTILISATION « GÉRER LES INFOS CLIENTS » - IDENTIFICATION DES FLUX EAI

Une pratique d'urbanisme courante consiste à utiliser une application comme référentiel des objets métier. Typiquement dans notre exemple, l'ERP SAP R/3 sert de référentiel des clients de l'entreprise, tandis que le CRM SIEBEL contient la référence des commandes. Bien entendu, ces deux applications ont besoin d'échanger leurs données car SIEBEL doit contenir les objets clients suivant son propre format pour fonctionner correctement et inversement pour les objets commandes.

Conformément aux propositions du profil « UML for EAI » de l'OMG, le diagramme d'activités ci-dessous peut être utilisé pour identifier puis étudier les échanges nécessaires. Ce diagramme décrit une partie du cas d'utilisation « Gérer les infos clients ».

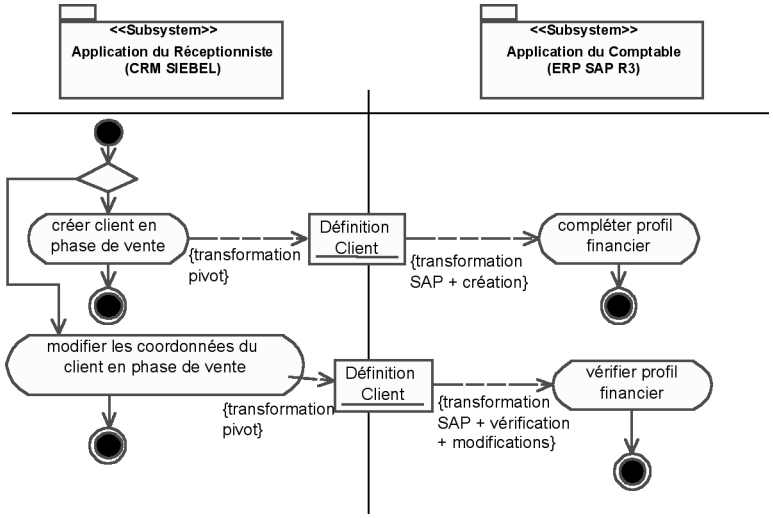


Figure 4-16. : Diagramme d'activités « Créer ou modifier un client en phase de vente »

Remarquez que les échanges s'expriment sous la forme de flux d'objets et que les contraintes servent à identifier les actions à réaliser dans le processus d'intégration. Nous voyons ainsi qu'il faut transformer la définition du client SIEBEL dans un format intermédiaire, dit pivot, puis transformer à nouveau ce format pour faciliter sa création dans SAP.

Identifier les classes candidates

Comme nous l'avons déjà mentionné, la définition des cas d'utilisation ne doit pas être une fin en soi !

La technique mise au point par Ivar Jacobson [Jacobson 92] comporte deux objectifs principaux :

- dialoguer avec le client sur son expression préliminaire de besoins grâce à une description fonctionnelle qu'il comprend facilement,
- préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse

Les paragraphes précédents de ce chapitre traitent du premier objectif. Nous allons maintenant aborder le second. Le travail que l'analyste va désormais effectuer complète d'ailleurs le précédent, car en mettant à jour les principales abstractions du système sous forme d'objets et de classes, l'analyste continue son dialogue avec le client. Il essaie ainsi d'obtenir rapidement un consensus sur les définitions des concepts-clés.

Les premières classes candidates identifiées dans cette phase doivent être des concepts connus des utilisateurs du système, ce qu'on appelle couramment (et abusivement, puisque ce sont des classes) des objets métier. Exemples pour l'étude de cas : *Mission*, *Commande*, *Client*, *Agence*, *Colis*, etc.

L'analyste ajoutera dans un second temps des concepts « applicatifs », liés à l'informatisation. Exemples pour l'étude de cas : *Etiquette à code-barre*, *Profil utilisateur*, etc.

Cherchez les noms communs importants dans les descriptions textuelles des cas d'utilisation. Vérifiez les propriétés « objet » de chaque concept (identité, propriétés, comportement), puis définissez ses responsabilités.

QU'EST-CE QU'UNE RESPONSABILITÉ ?

Une responsabilité est une sorte de contrat, ou d'obligation, pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. En fait, on peut dire que les attributs, les opérations, et les associations représentent les propriétés élémentaires qui contribueront à remplir les responsabilités de la classe.

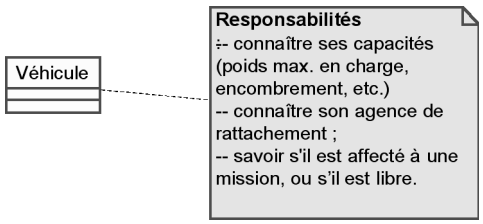


Figure 4-17. : Responsabilités de la classe Véhicule

En pratique, une classe doit avoir au moins une responsabilité, mais surtout un nombre très limité. Si elle comporte plus de cinq responsabilités, elle doit être subdivisée en plusieurs classes.

Graphiquement, les responsabilités peuvent être dessinées dans un compartiment séparé, au dessous des compartiments des attributs et des opérations. Dans la pratique, il s'agit de phrases courtes répertoriées dans une note graphique attachée à la classe.

Dans l'exemple précédent, il est probable que la première responsabilité se traduira par des attributs, et les suivantes par des associations.

On formalise ensuite ces concepts métier sous forme de classes et d'associations rassemblées dans un diagramme statique pour chaque cas d'utilisation. Ces diagrammes préliminaires, que nous appelons « diagramme de classes participantes », n'ont pas d'objectif d'exhaustivité. Ils servent uniquement à démarrer la découverte des classes du modèle d'analyse pour la partie de l'application délimitée par un cas d'utilisation. La réunion de tous les diagrammes, après élimination des classes et associations redondantes, doit représenter le squelette du modèle statique d'analyse.

ÉTUDE DE CAS : DIAGRAMME DES CLASSES PARTICIPANTES DU CAS D'UTILISATION « PLANIFIER UNE MISSION »

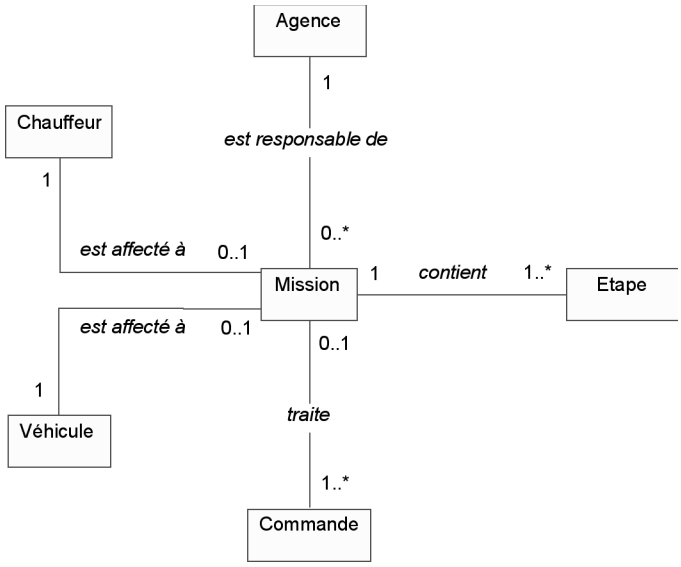


Figure 4-18. : Diagramme des classes participantes de « Planifier une mission »

Si l'on exploite la description textuelle du cas d'utilisation donnée plus haut, on peut ajouter en particulier :

- le type de mission : enlèvement, livraison, ou traction. Une association doit être ajoutée pour les missions de traction qui ont une agence de destination ;
- les bordereaux pour les missions validées : une feuille de route et une fiche par commande ;
- une mission est composée de différentes étapes. Une étape est planifiée pour chaque commande.

Il serait illusoire de penser figer les multiplicités des associations à ce moment-là. Des questions vont certainement se reposer régulièrement au fur et à mesure de l'avancement de l'analyse : 0 ou 1 ?, 1 ou plus ? C'est normal !

Remarque : nous avons volontairement limité l'utilisation des constructions du diagramme de classes, afin de ne pas anticiper sur les concepts qui seront abordés aux chapitres suivants.

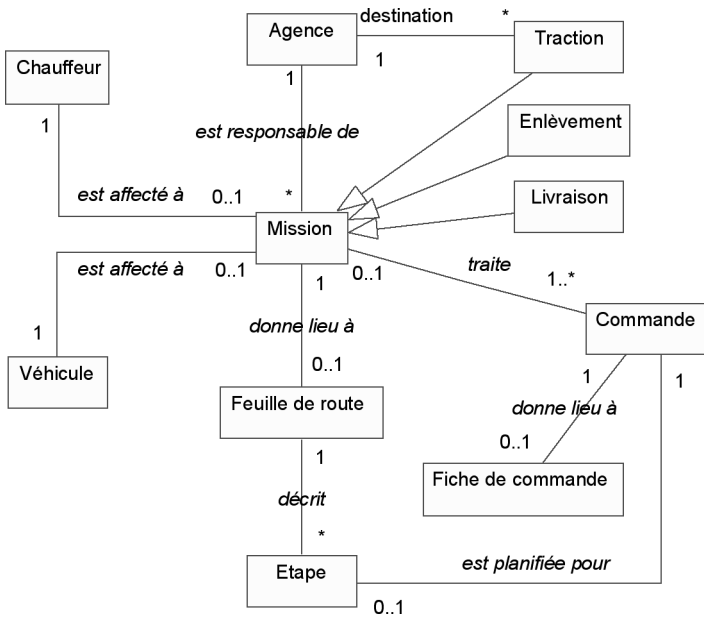


Figure 4-19. : Diagramme des classes participantes complété de « Planifier une mission »

ÉTUDE DE CAS : DIAGRAMME DES CLASSES PARTICIPANTES DU CAS D'UTILISATION « TRAITER UNE COMMANDE »

D'après la description initiale des besoins, on identifie les classes et associations suivantes :

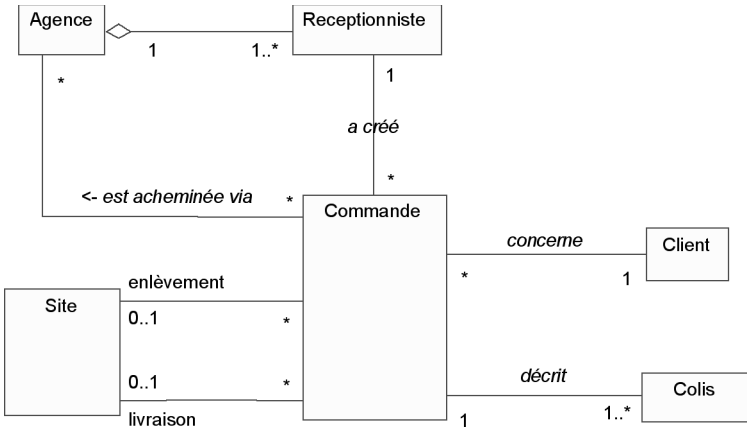


Figure 4-20. : Diagramme des classes participantes de « Traiter une commande »

Valider et consolider

La révision des cas d'utilisation doit absolument inclure une phase de présentation aux futurs utilisateurs et poser les questions-clés ci-après :

- Les frontières du système sont-elles bien définies ?
- Les acteurs sont-ils tous pris en compte (au moins une fois) ?
- Chaque cas d'utilisation a-t-il un processus de déclenchement (par un acteur) ?
- Le niveau d'abstraction des cas d'utilisation est-il homogène ?
- Toutes les fonctionnalités du système sont-elles traitées ?


Conseil

IL FAUT ASSURER LA TRAÇABILITÉ DES CAS D'UTILISATION AVEC L'EXPRESSION DES BESOINS !

Concrètement, on réalisera une matrice de traçabilité entre cas d'utilisation et éléments de cahier des charges. Cette matrice pourra être détaillée pour faire apparaître les différents enchaînements de chaque cas d'utilisation.

		Exigence 1	Exigence 2	Exigence 3	Exigence 4	Exigence 5
Use case A	scénario 1					
	scénario 2		x		x	x
	scénario 3		x		x	x
Use case B	scénario 1					
	scénario 2		x		x	
	scénario 3		x		x	
	scénario 4					
Use case C	scénario 1					
	scénario 2		x		x	

Cela permettra, dans la suite de l'analyse, d'établir le suivi des classes avec le cahier des charges, par le biais des cas d'utilisation ; il sera même possible de retrouver les opérations impliquées. L'aide d'un outil est alors précieuse. La figure suivante montre les informations que Rational/Rose sait fournir.

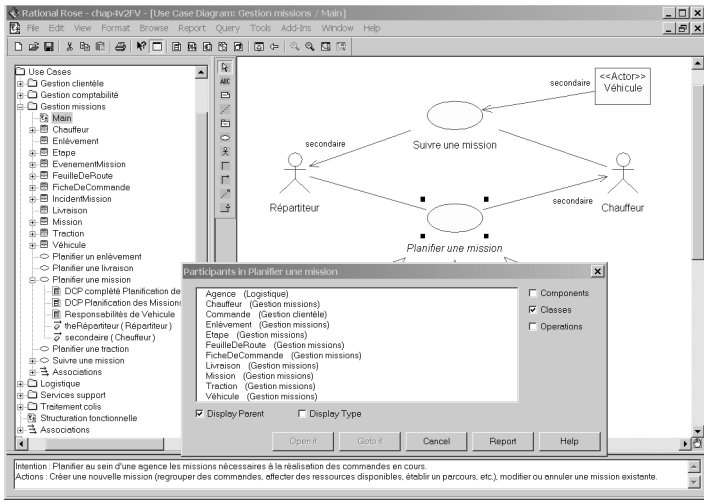


Figure 4-21. : Rational/Rose : « Report - Show Participants in UC »

Cette traçabilité entre besoins de haut niveau et opérations permet d'améliorer notablement la capacité de maintenance et d'évolution du code, tout au long de la vie de l'application.

SERVEZ-VOUS DES CAS D'UTILISATION POUR DÉFINIR VOS ITÉRATIONS !

Dans le cadre d'un développement itératif et incrémental, il est très utile de recourir au découpage en cas d'utilisation pour définir les itérations. À cet effet, il convient en premier lieu d'identifier les cas d'utilisation les plus critiques en termes de gestion des risques. Ces cas d'utilisation devront être traités prioritairement afin de lever au plus tôt les risques majeurs. Il sera également demandé au client d'affecter une priorité fonctionnelle à chaque cas d'utilisation, afin de livrer d'abord les cas d'utilisation les plus demandés. Ces deux critères pouvant être contradictoires, la décision du découpage en itérations incombe au chef de projet, qui doit le faire valider par le client.

Il faut aussi prendre en compte les éventuelles relations entre cas d'utilisation :

- développer plutôt les cas factorisés (<<include>>) avant ceux qui les utilisent,
- développer plutôt les cas qui étendent (<<extend>>) après les cas de base

ÉTUDE DE CAS : DÉFINITION DES ITÉRATIONS

Cas d'utilisation	Risque	Priorité	Itération
Traiter une commande	Moyen	Moyenne	5
Gérer les infos clients	Bas	Moyenne	6
Consulter les en-cours	Haut	Moyenne	3
Gérer la facturation	Bas	Basse	8
Suivre les règlements	Bas	Basse	8
Planifier une mission	Haut	Haute	2
Suivre une mission	Moyen	Haute	4
Réaliser l'inventaire	Bas	Moyenne	7
Manipuler les colis	Bas	Moyenne	7
Définir le plan de transport	Haut	Haute	1
Gérer les ressources	Moyen	Haute	3
Gérer les profils	Bas	Moyenne	9
S'authentifier	Bas	Moyenne	9

Tableau 4-5 : Définition des itérations par classement des cas d'utilisation.

À chaque cas d'utilisation de SIVEx, nous avons affecté :

- un risque (haut, moyen, bas),
- une priorité fonctionnelle (haute, moyenne, basse).

En fonction de ces informations et des dépendances entre cas d'utilisation, nous donnons un exemple de découpage du projet en itérations. Cette planification est évidemment donnée à titre illustratif, le processus 2TUP étant fondamentalement adaptatif et non pas prédictif.

Phases de réalisation de l'analyse des besoins fonctionnels

L'analyse des besoins fonctionnels a pour objectifs principaux de :

- compléter le recueil initial des besoins effectué pendant l'étude préliminaire : nous avons expliqué comment utiliser à cet effet le concept central de cas d'utilisation proposé par UML ;

- préparer l'analyse orientée objet : pour chaque cas d'utilisation, nous avons vu comment identifier les classes candidates du modèle statique d'analyse.

La démarche mise en œuvre dans ce chapitre est synthétisée par la figure suivante :

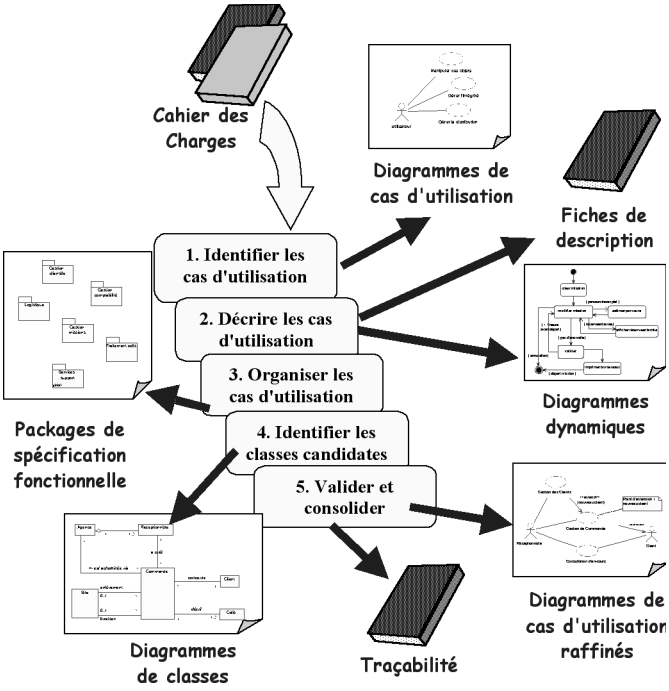
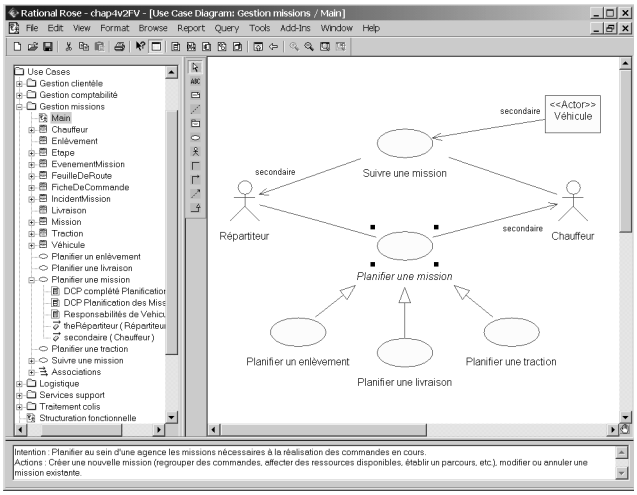


Figure 4-22. : Démarche de capture des besoins fonctionnels

Sur le site Web (www.editions-eyrolles.com)

Pour consulter le modèle UML contenant les cas d'utilisation et les diagrammes associés, ouvrez le fichier *chap4.mdl* (au format Rational/Rose)


Rational Rose - chap4v2FV - [Use Case Diagram: Gestion missions / Main]
File Edit View Format Browse Report Query Tools Add-Ins Window Help
Use Cases
Gestion clientèle
Gestion comptabilité
Gestion missions
Main
Chauffeur
Enlèvement
Etape
EvenementMission
FeuilleDeRoute
FicheDeCommande
IncidentMission
Livraison
Mission
Traction
Véhicule
Planifier un enlèvement
Planifier une livraison
Planifier une mission
DCP complété Planification
DCP Planification des Miss
Responsabilités de Vehicu
theRépartiteur (Répartiteur
secondaire (Chauffeur)
Planifier une traction
Suivre une mission
Associations
Logistique
Services support
Traitement colis
Structuration fonctionnelle
<<Actor>>
Véhicule
secondaire
Suivre une mission
secondaire
Répartiteur
secondaire
Chauffeur
Planifier une mission
Planifier un enlèvement
Planifier une livraison
Planifier une traction
Intention : Planifier au sein d'une agence les missions nécessaires à la réalisation des commandes en cours.
Actions : Créer une nouvelle mission (regrouper des commandes, affecter des ressources disponibles, établir un parcours, etc.), modifier ou annuler une mission existante.