

Chapitre VI :

Diagramme d'interaction

6.1 Introduction

Les deux diagrammes les plus utilisés pour spécifier les interactions entre les différentes parties du système sont : le diagramme de séquence et le diagramme de communications. Il est important de signaler ici que les diagrammes de séquence et de communications mettent en œuvre principalement des objets et non des classes, ils représentent une vue dynamique du système.

Le diagramme de séquence montre l'ordre des échanges de messages et le passage du temps. C'est un diagramme dit temporel. Par contre, le diagramme de communications met moins en évidence l'aspect temporel mais en faisant ressortir les relations entre objets. Les principaux concepts sont les objets participants à l'échange des messages, le temps, les messages, et la création et la suppression de participants. Comme ces diagrammes deviennent vite imposants en taille, la notion de fragment permet de les construire de façon modulaire. Un diagramme de séquence spécifie également le comportement d'un cas d'utilisation ou d'une partie de celui-ci. Notons que le rôle du diagramme de communication est de montrer les différents messages qui se propagent d'un objet à l'autre : il attire l'attention de l'analyste et du concepteur sur le fait qu'un message transite sur une association, qui doit donc exister entre la classe de l'objet appelant et la classe de l'objet appelé. Comme dans les diagrammes de séquence, un objet doit avoir une méthode appropriée pour traiter chaque événement qu'il reçoit.

6.2 Éléments graphiques d'un diagramme d'interaction

Les principales informations contenues dans un diagramme d'interaction sont les messages échangés entre les lignes de vie. Un message définit une communication particulière entre des lignes de vie. Les messages les plus communs sont :

- Envoi d'un signal
- L'invocation d'une opération
- La création ou la destruction d'une instance

Pour UML, l'envoi d'un signal ou l'invocation d'une opération sont deux sortes de messages qui se représentent de la même façon. Par contre, UML fait la différence entre un message synchrone et un message asynchrone.

Exemple d'application :

Nous étudions le système d'information d'un garage (Bauchet 2015) et nous illustrons le diagramme de séquence sur le cas d'utilisation « créer un client ». Le cadre, étiqueté « creer_utilisateur », est ce qu'on appelle un fragment : il permet de nommer et délimiter un diagramme de séquence. A la différence d'un diagramme de classes (qui est générique à toutes les situations), le diagramme de séquence (comme le diagramme d'objets) décrit une situation précise : ici, nous décrivons la création d'un nouveau client (ID2457) par un garagiste donné (Robert, qui utilisera le système). Le diagramme de séquence insiste sur les interactions entre les acteurs et les objets (les interactions étant dans la majorité des cas initialisées par un acteur).

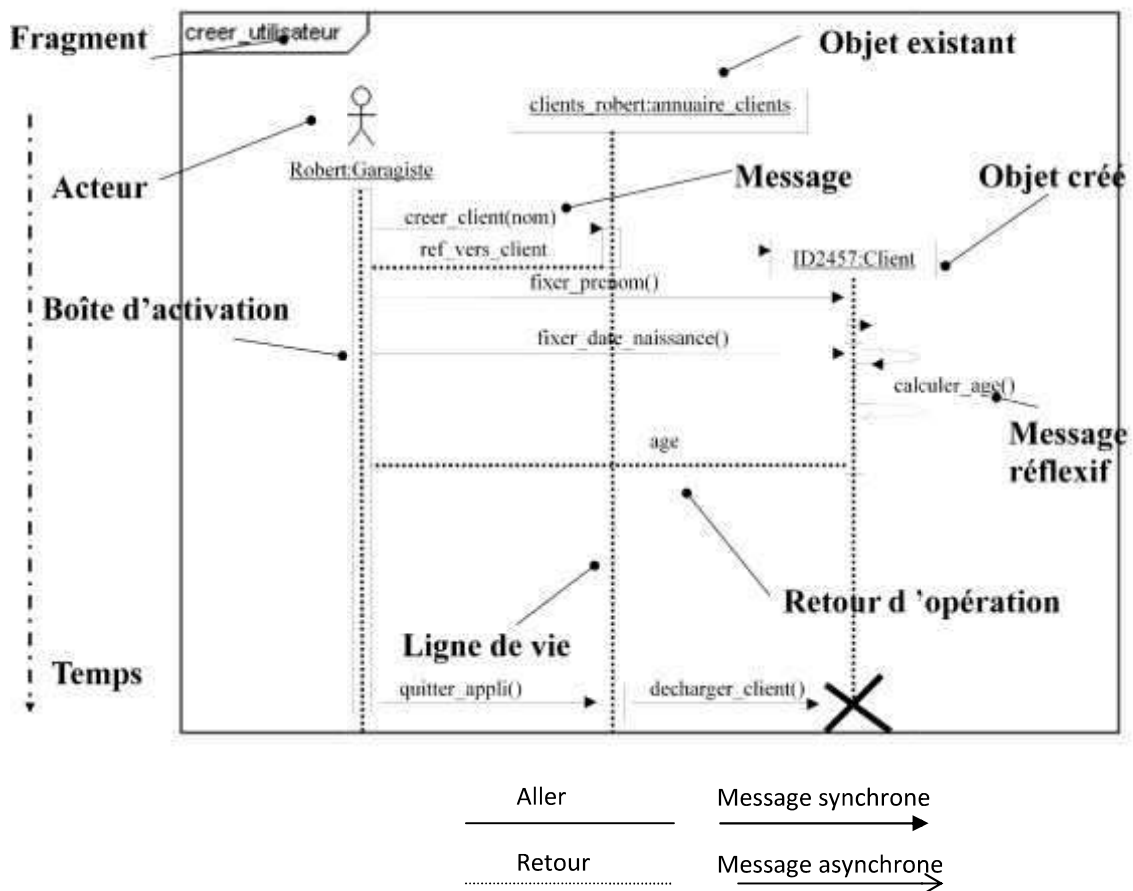


Figure 6.1 Diagramme de séquence : Notation.

Ce diagramme représente la sémantique de chacun des éléments graphiques mis à disposition de l'analyste pour la documentation de la dynamique d'un domaine (et du concepteur pour décrire la dynamique des traitements).

Chaque objet (et acteur) dispose d'une travée (d'un couloir, un peu comme dans le diagramme d'activité).

Dans le diagramme, trois éléments interagissent :

- l'acteur ;
- l'objet clients_robort de la classe annuaire_clients ;
- l'objet id2457 de la classe client.

Les objets qui apparaissent dans le diagramme de séquence doivent nécessairement être des instances de classes

- qui, en analyse, existent dans le modèle du domaine (diagramme de classes d'analyse) ;
- qui, en conception, existent dans le modèle de conception (diagramme de classes de conception).

Le (voire les) acteur(s) ne sont pas représentés par une classe, l'acteur Robert:Garagiste n'apparaît pas dans le diagramme d'objets, ni dans le diagramme de classes sous la forme de la classe Garagiste. Dans le diagramme d'interaction, l'acteur représente l'être humain qui interagit avec le système. Notons que par défaut, l'appel est synchrone. Avant d'envoyer un message, il faut que l'objet client soit créé dans la figure ci-dessus (l'opération fixer_prenom ne peut être exécutée avant le retour ref_vers_client, n'oublions pas que l'axe vertical descendant représente le temps qui s'écoule).

Contrairement à un diagramme de séquence, un diagramme de communication rend compte de l'organisation spatiale des participants à l'interaction, il est souvent utilisé pour aider à valider les associations du diagramme de classes en les utilisant comme support de transmission des messages.

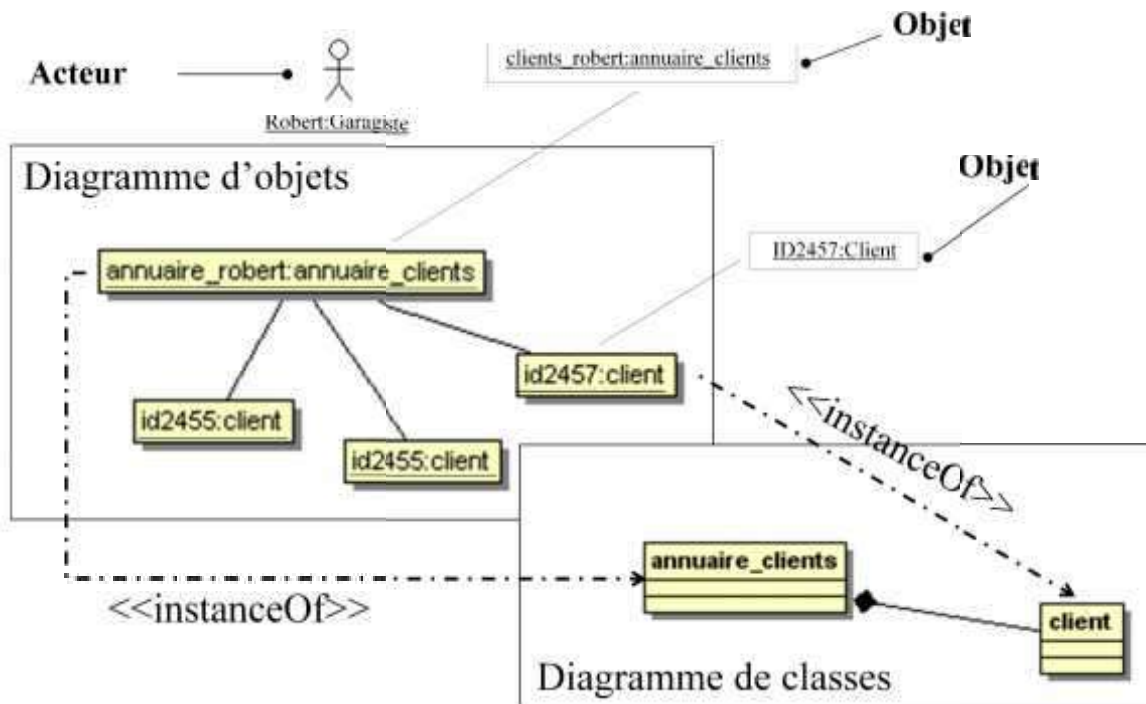


Figure 6.2.A Diagramme de communication : Notation.

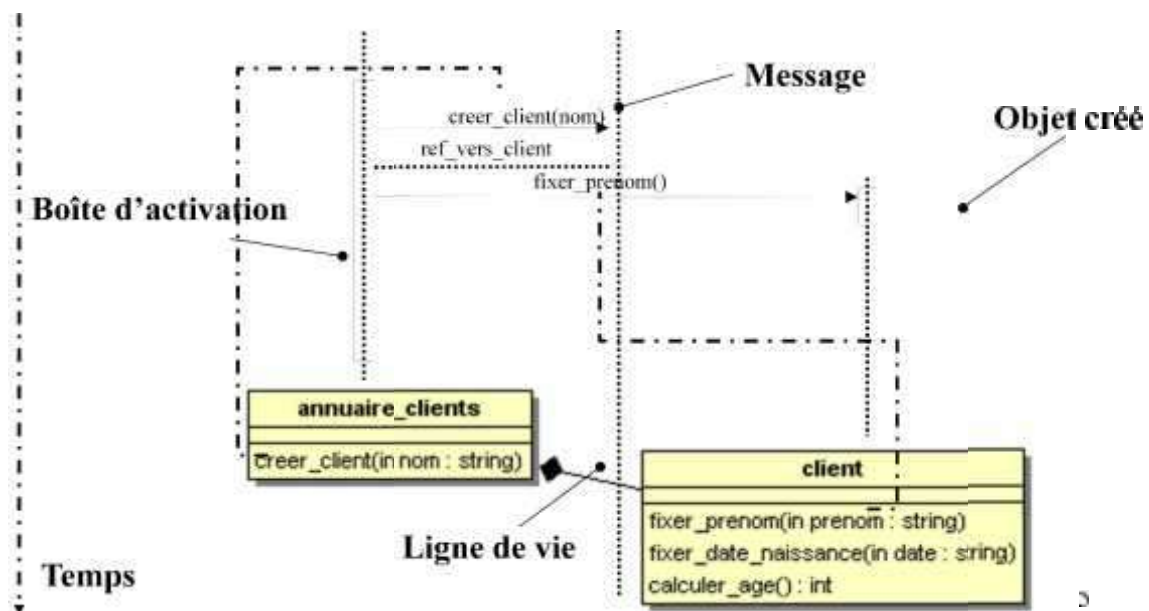


Figure 6.2.B diagramme de séquence et de classe servant à créer le diagramme de communication.

A priori, sur un diagramme de classes, on représente des classes, dans un diagramme d'objets, on représente des objets. Il est cependant possible de réunir sur un même diagramme Classes et Objets: la relation d'instanciation entre les deux étant représentée par la dépendance « `instanceOf` ».

Un message est envoyé par un objet (ou acteur) vivant vers un objet vivant (on observe la ligne de vie de part et d'autre du message). Pour envoyer un message, un objet doit avoir une fonction activée (boîte d'activation).

Lorsque le message « `creer_client` » (envoyé par l'acteur Robert) arrive à l'objet destination (l'objet `clients_robert` de type ou classe `annuaire_clients`), ce dernier déclenche l'appel à l'opération « `creer_client` » qui doit exister sur la classe `annuaire_clients` (dont l'objet `clients_robert` est une instance).

Il est à noter que, pour envoyer un message, un objet doit disposer une référence vers objet récepteur du message (dans le cas d'un acteur, cela signifie qu'il existe sur l'interface utilisateur un élément graphique lui permettant de référencer l'annuaire). On note d'ailleurs que l'opération retourne une valeur (notée `ref_vers_client`) qui sera justement utilisée immédiatement par l'acteur pour faire appel à une opération (en l'occurrence « `fixer_prenom` ») de l'objet « `ID2457` » (troisième ligne de vie).

6.3 Formalisme d'un diagramme d'interaction

Les diagrammes d'interaction sont utilisés tout au long du cycle de vie d'un projet (depuis le recueil de besoin jusqu'à la phase de conception). Ils servent à décrire le cas d'utilisation dans la phase en amont, à modéliser la mise en œuvre d'une classe ou d'une opération d'une classe et plus généralement, à ajouter un aspect dynamique à la modélisation d'un système.

6.3.1 Construction d'un diagramme de séquence

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps y est représenté explicitement par une dimension (la dimension verticale) et s'écoule de haut en bas (cf. figure 6.3).

On peut facilement remarquer dans les figures (6.1, 6.3) que l'objet `ID2457` n'était pas situé sur la première ligne (tout en haut) comme l'objet `clients_Robert`. Cela signifie qu'au démarrage de la séquence, l'objet `ID2457` n'existait pas. Il est créé à l'instant correspondant à la ligne 2.

Enfin, la croix représente la fin de vie d'un objet : au-delà de la croix, la ligne de vie pointillée n'est plus représentée.

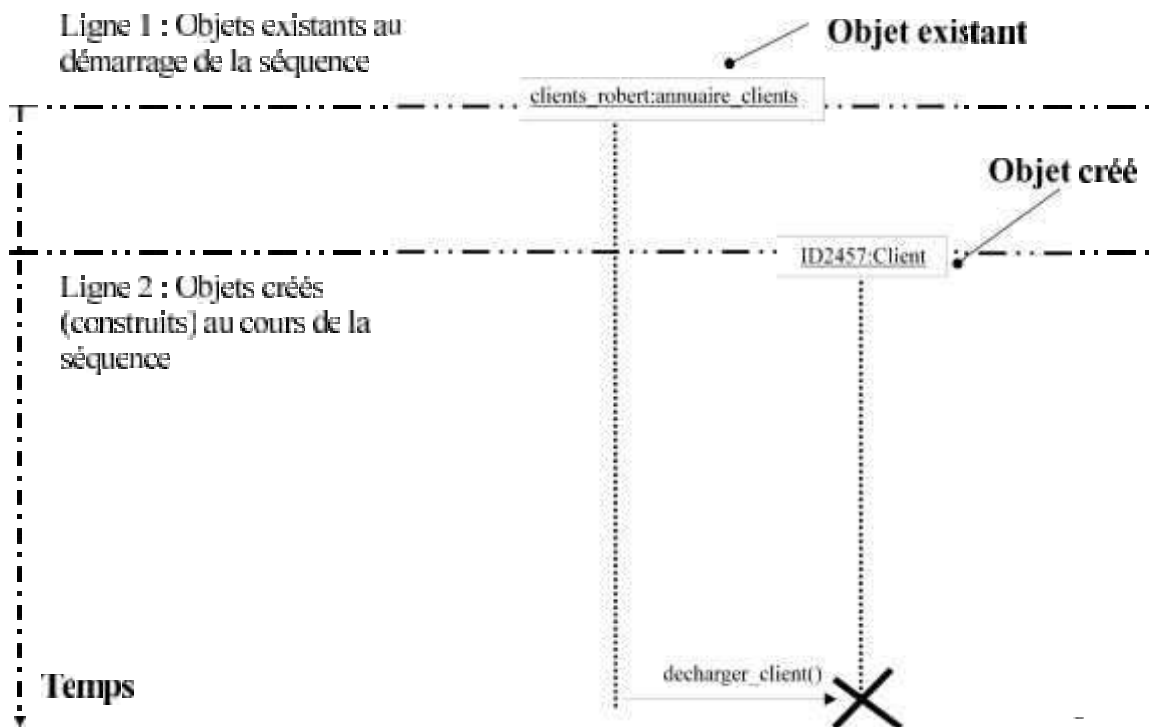


Figure 6.3 fonctionnement d'un diagramme de séquence.

6.3.1.1 Fragments dans un diagramme de séquence

Le diagramme (fragment intitulé « complet ») indique que s'enchaînent les séquences référencées par les fragments « initialiser_systeme », « créer_client », ... UML offre un formalisme pour exprimer l'emboîtement. Pour simplifier un diagramme et insister sur un enchaînement particulier, il est possible de considérer que les objets existants ont été créés dans d'autres diagrammes. Donc on n'est pas obligé d'ajouter les fragments contenant la création d'objet.

Le fragment (intitulé « creer_client ») est un diagramme de séquence système, (seule les interactions avec le système sont décrites, on ne décrit pas ce qui se passe dans le système). Quand on souhaite élaborer un diagramme de séquence développé, on commence en général par élaborer le DSS, puis on cherche, pour chaque message, quels objets, instances de classes prises dans le modèle du domaine (diagramme de classes d'analyse), sont utilisés dans le traitement de cette interaction.

Le Diagramme de Séquence Système (DSS) est équivalent à la partie « Actions » du formulaire de Cockburn (Bauchet 2015). A noter que la terminologie entre le formulaire de Cockburn, très proche de l'utilisateur final, peut être simplifiée comme suit :

Gargiste	Système
1-Ouvrir une fiche client dans l'annuaire client	2-attribuer le dernier numéro libre
3-Saisir le nom	
4-Saisir le prénom	
5-Saisir la date de naissance	6-retourner l'age client
7-Refermer l'annuaire client	

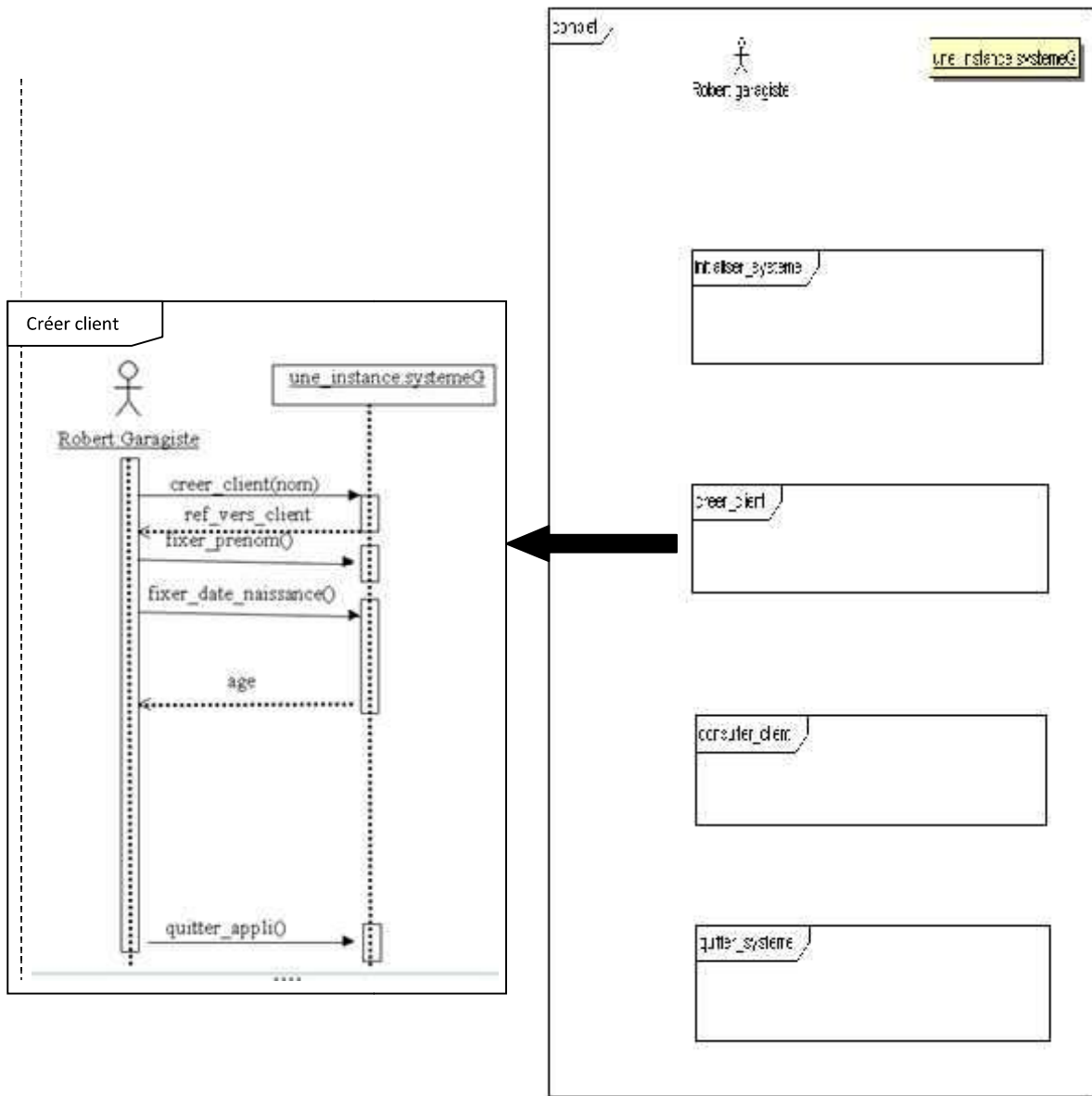


Figure 6.4 Fragments dans le diagramme de séquence.

Les mots clés pouvant être utilisés pour définir des fragments d'un diagramme de séquence sont les suivants :

- alt** : fragment alternatif, conditions dans les gardes
- loop** : fragment à répéter tant que la condition de garde est vraie. Il représente aussi la notion de boîte d'action avec itérateur.
- opt** : fragment optionnel exécuté si la garde est vraie
- par** : fragments qui s'exécutent en parallèle
- region** : region critique dans laquelle un seul thread doit s'exécuter
- ref** : passage à un autre diagramme de séquence

Nous présentons dans la figure 6.5 les fragments alt et loop :

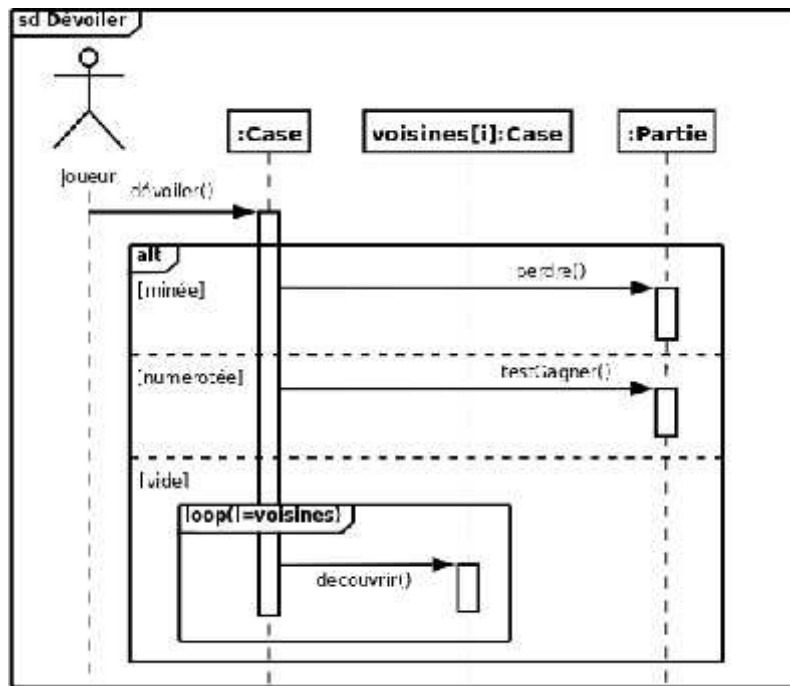


Figure 6.5 Diagramme de séquence illustrant le dévoilement d'une case au jeu du démineur (Audibert 2009).

Le but du jeu « démineur » est de découvrir toutes les cases libres sans faire exploser les mines. La partie est gagnée si toutes les cases sauf les mines sont découvertes, elle est perdue si le joueur tombe sur une mine. Le chiffre qui s'affiche sur les cases cliquées indique le nombre de mines se trouvant à proximité : à gauche ou à droite, en haut ou en bas, ou en diagonale. Le fait que la case est numérotée on peut savoir lesquelles des cases voisines contenant une mine. Le fait de cliquer sur une case contenant une mine cela signifie une perte.

Le fragment alt joue le rôle d'une exécution à choix multiple (condition *switch*). Chaque opérande détient une condition de garde. L'absence de condition de garde implique une condition vraie (*true*). La condition *else* est vraie si aucune autre condition n'est vraie. Le fragment loop représente une itération : c'est-à-dire le joueur refait la même action plusieurs fois jusqu'à finir le jeu sans exploser aucune mine.

6.3.2 Construction d'un diagramme de communication

La construction d'un diagramme de communication (nommé diagramme de collaboration dans UML1) peut servir à tester un modèle statique pendant la conception ; il s'agit de représenter un scénario dans lequel les classes du diagramme de classes sont instanciées pour créer les objets nécessaires à l'exécution du scénario. Il complète le diagramme de classes qui représente la structure statique d'un système en spécifiant le comportement des classes et interfaces ainsi que l'utilisation possible de leurs opérations.

Le diagramme de communication est considéré comme une version simplifiée d'un diagramme de séquence. On l'utilise dans le but de clarifier le domaine d'étude pour représenter la collaboration entre le domaine d'étude et les partenaires. Il permet de déterminer les flux entrants, les flux sortants et les acteurs externes.

Un scénario de diagramme de communication utilise trois types de concepts :

- des instances des classes (objets du système étudié) qui interviennent lors du scénario ;
- des acteurs qui reçoivent ou émettent des événements lors du scénario ;
- les appels aux méthodes ou flux d'information vers un acteur externe.

Les scénarios représentent les exécutions du système. Ils sont très utiles pour les phases de tests et de maintenance. Un diagramme de communication est associé à un cas d'utilisation ; il est à noter qu'à un diagramme de cas d'utilisation on peut associer plusieurs diagrammes de communication. Chaque diagramme de communication correspondant à une exécution particulière. La numérotation des flux indique l'ordre chronologique et aussi la relation de causalité.

Dans un diagramme de communication, les messages sont généralement ordonnés selon un numéro de séquence croissant (voir figure ci-dessous).

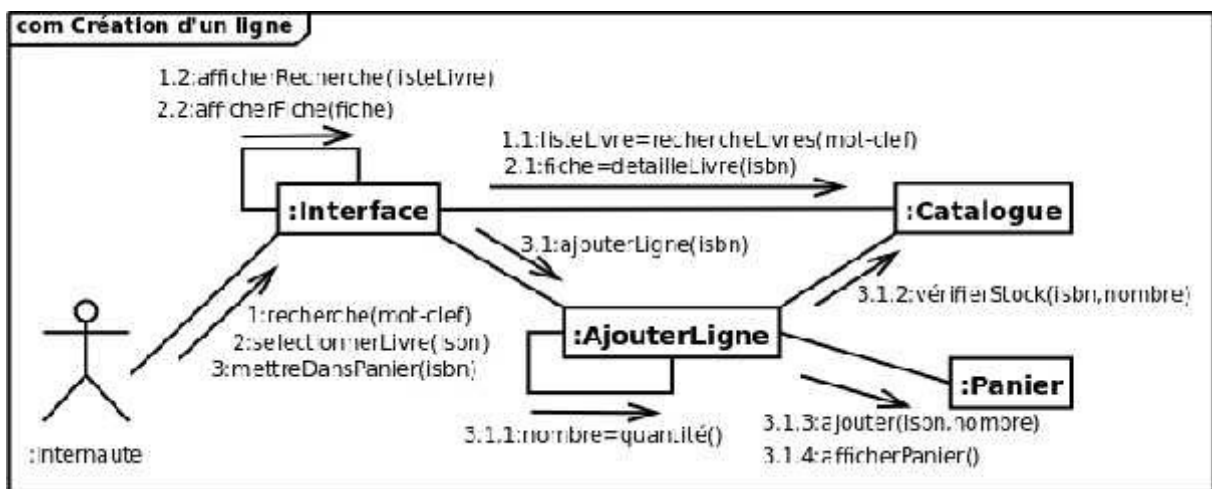


Figure 6.6 : Diagramme de communication illustrant la recherche puis l'ajout, dans son panier virtuel, d'un livre lors d'une commande sur Internet (Charroux et al 2005).

6.4 Diagramme de séquence et de communication : points forts et points faibles

La spécification UML est plus centrée sur les diagrammes de séquences que sur les diagrammes de communication.

- Davantage de réflexion et d'efforts ayant été consacrés à leur notation et leur sémantique.
- En conséquence, les outils les prennent mieux en charge et les options de notation sont plus nombreuses.

Les diagrammes de communication sont intéressants lorsqu'on applique UML « en mode esquisse » en modélisant au tableau parce qu'ils permettent beaucoup mieux d'exploiter l'espace.

- Possibilité d'ajouter/supprimer des boîtes n'importe où sur l'axe horizontal et vertical.
- Important vu la fréquence des changements intervenant lors de la conception.

Au contraire, dans les diagrammes de séquence, les nouveaux objets doivent être ajoutés à droite, ce qui impose des limites car le bord droit de la page (ou du tableau) est rapidement épuisé (tandis que de l'espace vertical est disponible).

- Les diagrammes de séquence permettent une meilleure visualisation des flots d'appels, parce qu'on les lit simplement de haut en bas.
- Les diagrammes de séquence sont excellents pour les besoins de la documentation ou pour suivre facilement le flot d'appels généré automatiquement par la rétro ingénierie.
- Les diagrammes de communication ne permettent quant à eux de retrouver la séquence des appels qu'en se référant à leur numérotation, comme « 1 », « 2 »,... .

6.5 Conclusion

Les diagrammes de séquence et de communication montrent l'aspect dynamique d'un système. Ce ne sont pas les seuls : les diagrammes d'activité et diagramme d'états-transitions, qui sera présenté au chapitre suivant, décrivent eux aussi l'aspect dynamique du système. A la différence des diagrammes d'activité et d'états-transitions qui représentent la dynamique du système en détaillant les traitements en phase de réalisation, les diagrammes de séquence et de communication sont mieux appropriés pour la description des interactions entre les éléments d'un système.

Diagramme de séquence et diagramme de communication peuvent décrire une même interaction mais de façons différentes. Le concepteur a le choix entre l'utilisation du diagramme de séquence qui montre le séquençement temporel des messages et l'utilisation d'un diagramme de communication qui décrit la structure spatiale des participants à une interaction.

Pour compléter les diagrammes servant à montrer l'aspect dynamique du système, nous présentons dans le chapitre suivant le diagramme d'états-transitions. Ce diagramme est utilisé avant tout pour décrire le cycle de vie des objets d'un système.

