

Chapitre 5

Modélisation dynamique : étude de cas

Mots-clés

■ Acteur ■ Cas d'utilisation, scénario ■ Diagramme de séquence système ■ Diagramme de contexte dynamique ■ Message ■ Diagramme d'états ■ État, transition ■ Événement ■ Condition ■ Action - Activité.

Ce chapitre va nous permettre d'illustrer pas à pas, à partir d'une nouvelle étude de cas, les principaux concepts et diagrammes UML pour le point de vue dynamique.

En commençant par identifier les acteurs et les cas d'utilisation, nous dessinerons tout d'abord un diagramme de séquence « système ». Puis, nous réaliserons un diagramme de communication particulier (que nous appellerons diagramme de contexte dynamique) afin de répertorier tous les messages que les acteurs peuvent envoyer au système et recevoir.

Après ce travail préliminaire, nous nous embarquerons dans une description en profondeur de la dynamique souhaitée du système. Nous insisterons tout particulièrement sur le diagramme d'états qui est à notre avis trop souvent sous-employé, alors que c'est un diagramme extrêmement utile pour décrire avec précision des comportements complexes, et ce dès l'analyse.

En complément des concepts de base, nous expliquerons la bonne utilisation des concepts avancés tels que :

- Événement interne : *when*
- État composite et sous-états exclusifs
- Transition propre et transition interne
- Pseudo-état : *History*
- Envoi de message : *send*

PRINCIPES ET DÉFINITIONS DE BASE

DIAGRAMME D'ÉTATS

UML a repris le concept bien connu de *machine à états finis*, qui consiste à s'intéresser au cycle de vie d'une instance générique d'une classe particulière au fil de ses interactions avec le reste du monde, dans tous les cas possibles. Cette vue locale d'un objet, qui décrit comment il réagit à des événements en fonction de son état courant et comment il passe dans un nouvel état, est représentée graphiquement sous la forme d'un *diagramme d'états*¹.

Un diagramme d'états pour quelles classes ?

Toutes les classes du modèle statique ne requièrent pas nécessairement une machine à états, représentée par un diagramme d'états. Il s'agit donc de trouver celles qui ont un comportement dynamique complexe nécessitant une description poussée. Cela correspond à l'un des deux cas suivants :

- les objets de la classe peuvent-ils réagir différemment à l'occurrence du même événement ? Chaque type de réaction caractérise un état particulier ;
- la classe doit-elle organiser certaines opérations dans un ordre précis ? Dans ce cas, des états séquentiels permettent de préciser la chronologie forcée des événements d'activation.

Comment le construire ?

1. Représentez tout d'abord la séquence d'états qui décrit le comportement nominal d'un objet, avec les transitions qui y sont associées.
2. Ajoutez progressivement les transitions qui correspondent aux comportements alternatifs ou d'erreur.

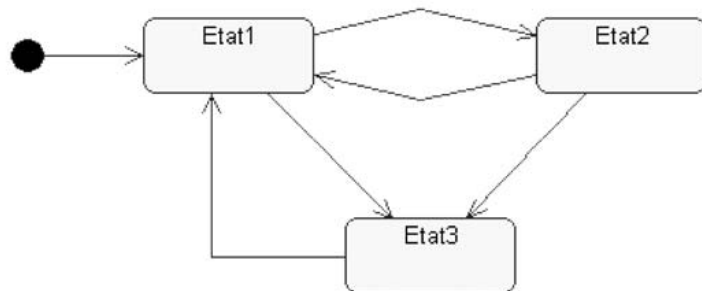
1. UML utilise beaucoup le terme *statechart*, issu des travaux de D. Harel (voir www.ilogix.com).

3. Complétez les effets sur les transitions et les activités dans les états.
4. Structurez le diagramme en sous-états s'il devient trop complexe.

Comment le représenter ?

Un diagramme d'états est un graphe orienté d'états et de transitions.

Figure 5-1.
*Notation de base du
diagramme d'états*



ÉTAT

Un *état* représente une situation durant la vie d'un objet pendant laquelle :

- il satisfait une certaine condition ;
- il exécute une certaine activité ;
- ou bien il attend un certain événement.

Un objet passe par une succession d'états durant son existence. Un état a une durée finie, variable selon la vie de l'objet, en particulier en fonction des événements qui lui arrivent.

Comment les identifier ?

Comment trouver les états d'une classe ? Pour faire un parallèle, on peut dire qu'il est aussi difficile de trouver les bons états dans le modèle dynamique que les bonnes classes dans le modèle statique ! Il n'y a donc pas de recette miracle, cependant trois démarches complémentaires peuvent être mises en œuvre :

- la recherche intuitive repose sur l'expertise métier. Certains états fondamentaux font partie du vocabulaire des experts du domaine et sont identifiables a priori ;
- l'étude des attributs et des associations de la classe peut donner des indications précieuses : cherchez des valeurs seuils d'attributs qui modifient la dynamique, ou des comportements qui sont induits par l'existence ou l'absence de certains liens ;
- une démarche systématique peut également être utilisée : classe par classe, cherchez le diagramme d'interaction le plus représentatif du comportement des instances de cette classe, associez un état à chaque intervalle entre événements émis ou reçus par

une instance et placez les transitions. Reproduisez ensuite cette démarche avec tous les scénarios faisant intervenir des instances de la classe, afin d'ajouter de nouvelles transitions ou de nouveaux états. La difficulté principale consiste à trouver ensuite les boucles dans le diagramme, afin de ne pas multiplier les états.

État initial et état final

En plus de la succession d'états « normaux » correspondant au cycle de vie d'un objet, le diagramme d'états comprend également deux pseudo-états :

- L'état initial du diagramme d'états correspond à la création de l'instance.
- L'état final du diagramme d'états correspond à la destruction de l'instance.

TRANSITION

Une transition décrit la réaction d'un objet lorsqu'un événement se produit (généralement l'objet change d'état). En règle générale, une transition possède un événement déclencheur, une condition de garde, un effet et un état cible.

ÉVÉNEMENT

En UML, un événement spécifie qu'il s'est passé quelque chose de significatif, localisé dans le temps et dans l'espace. Dans le contexte des machines à états finis, il représente l'occurrence d'un stimulus qui peut déclencher une transition entre états.

UML propose de distinguer quatre sortes d'événements :

- la *réception d'un message* envoyé par un autre objet, ou par un acteur. L'envoi d'un message est en général asynchrone ;
- l'*appel d'une opération (call event)* sur l'objet récepteur. L'événement d'appel est en général synchrone ;
- le *passage du temps (time event)*, qui se modélise en utilisant le mot-clé **after** suivi d'une expression représentant une durée, décomptée à partir de l'entrée dans l'état courant ;
- un *changement* dans la satisfaction d'une condition (*change event*). On utilise alors le mot-clé **when**, suivi d'une expression booléenne. L'événement de changement se produit lorsque la condition passe à vrai.

Un événement peut porter des paramètres qui matérialisent le flot d'informations ou de données entre objets.

MESSAGE

Un message est une transmission d'information unidirectionnelle entre deux objets, l'objet émetteur et l'objet récepteur.

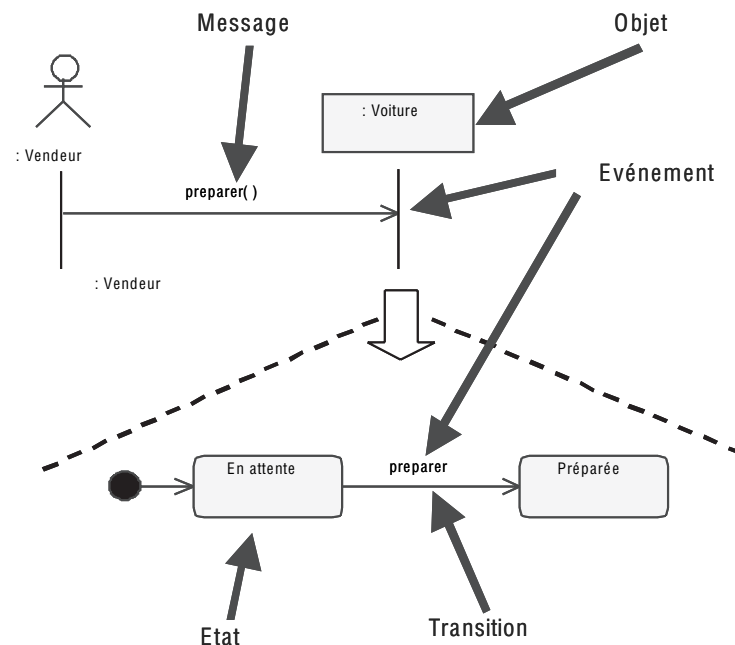
Le mode de communication peut être synchrone ou asynchrone. La réception d'un message est un événement qui doit être traité par le récepteur.

CONDITION

Une condition (ou condition de garde) est une expression booléenne qui doit être vraie lorsque l'événement arrive pour que la transition soit déclenchée. Elle peut concerner les attributs de l'objet ainsi que les paramètres de l'événement déclencheur. Plusieurs transitions avec le même événement doivent avoir des conditions de garde différentes.

Comment représenter les concepts dynamiques de base ?

Figure 5-2.
Concepts dynamiques de base



EFFET : ACTION OU ACTIVITÉ

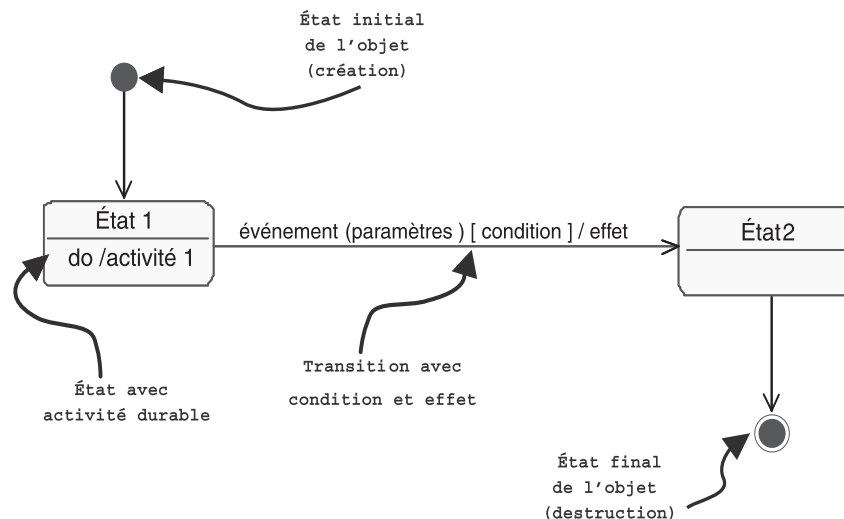
Une transition peut spécifier un comportement optionnel réalisé par l'objet lorsque la transition est déclenchée. Ce comportement est appelé « *effet* » en UML 2 : cela peut être une simple action ou une séquence d'actions (une activité). Une action élémentaire² peut représenter la mise à jour d'un attribut, un appel d'opération, la création ou la destruction d'un autre objet, ainsi que l'envoi d'un signal à un autre objet. Les *activités* associées aux transitions sont considérées comme atomiques, c'est-à-dire qu'elles ne peuvent être interrompues. Ce sont typiquement des séquences d'actions.

2. UML 2 a apporté la définition de plus de cinquante actions élémentaires avec leur nom et leur sémantique.

Les *activités durables* (*do-activity*), au contraire, ont une certaine durée, sont interruptibles et sont donc associées aux états.

COMMENT REPRÉSENTER LES DIAGRAMMES D'ÉTATS ?

Figure 5-3.
*Notation plus complète
du diagramme d'états*



ÉTUDE D'UN PUBLIPHONE À PIÈCES

Cette étude de cas concerne un système simplifié de Publiphone à pièces.

1. Le prix minimal d'une communication interurbaine est de 0,2 euros.
2. Après l'introduction de la monnaie, l'utilisateur a 2 minutes pour composer son numéro (ce délai est décompté par le standard³).
3. La ligne peut être libre ou occupée.
4. Le correspondant peut raccrocher le premier.
5. Le Publiphone consomme de l'argent dès que l'appelé décroche et à chaque unité de temps (UT) générée par le standard.
6. On peut ajouter des pièces à tout moment.
7. Lors du raccrochage, le solde de monnaie est rendu.

À partir de ces six phrases, nous allons progressivement :

1. Identifier les acteurs et les cas d'utilisation.



3. Nous utilisons le terme « standard », mais il représente en fait le réseau téléphonique dans son ensemble.

2. Construire un diagramme de séquence système.
3. Construire le diagramme de contexte dynamique.
4. Élaborer le diagramme d'états du Publiphone.

Étape 1 – Identification des acteurs et des cas d'utilisation

En premier lieu, nous allons identifier les acteurs et les cas d'utilisation du Publiphone à pièces.



EXERCICE 5-1.

Diagramme de cas d'utilisation

Dessinez le diagramme de cas d'utilisation du Publiphone à pièces.

solution

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

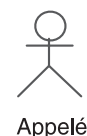
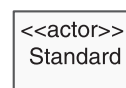
Si nous procédons à une analyse linguistique de l'énoncé, nous obtenons les cinq candidats suivants : utilisateur, standard, correspondant, Publiphone, appelé.

Éliminons tout de suite Publiphone puisqu'il s'agit du système lui-même. En revanche, le standard est bien un acteur (non-humain) connecté directement au système.

La seule difficulté concerne les acteurs humains : utilisateur, correspondant et appelé. Comme les deux derniers termes semblent être synonymes, nous pouvons garder le mot *appelé* et renommer, par souci de symétrie, utilisateur en *appelant*.

Figure 5-4.

Liste préliminaire des acteurs

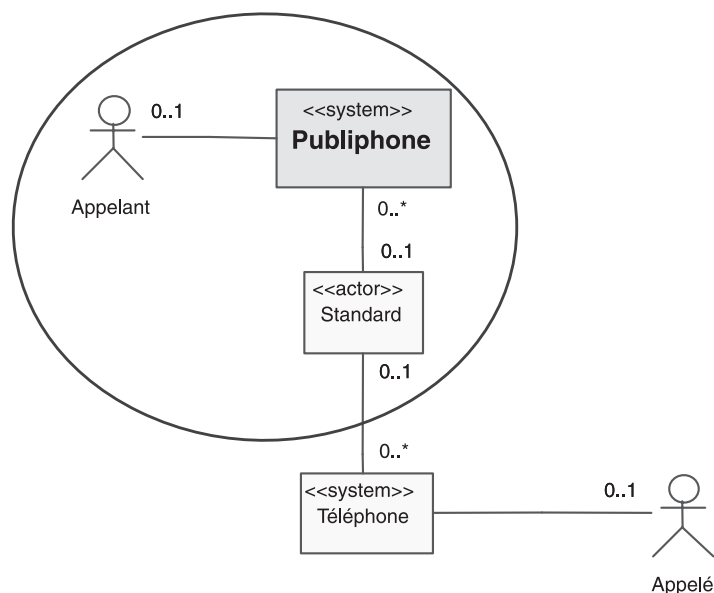


Comment les acteurs utilisent-ils le Publiphone ? La seule utilisation vraiment intéressante dans notre contexte est celle de l'appelant qui téléphone à l'appelé. Le standard sert en fait d'intermédiaire entre les deux. Si nous affinions encore notre analyse, nous nous apercevons rapidement que l'appelé n'interagit pas directement avec le Publiphone : il est complètement masqué par le standard.

Une illustration graphique de ce problème de détermination de frontière est donnée sur le diagramme de contexte statique suivant.

Figure 5-5.

Diagramme de contexte statique élargi du Publiphone

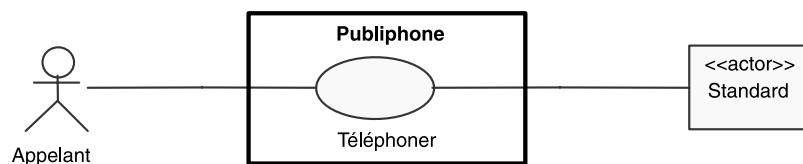


On voit bien sur le diagramme précédent que l'appelant communique avec l'appelé par le biais de trois systèmes connectés : le Publiphone, le standard et le téléphone de l'appelé. On notera la symétrie du schéma en regard du standard, qui joue le rôle d'acteur par rapport aux deux autres systèmes de même nature.

L'appelé est donc un acteur indirect par rapport au Publiphone. Nous ne le retiendrons pas pour notre diagramme de cas d'utilisation qui est finalement très simple.

Figure 5-6.

Diagramme de cas d'utilisation du Publiphone



Étape 2 – Réalisation du diagramme de séquence système

Avant de plonger dans les arcanes du diagramme d'états du Publiphone, nous allons le préparer en réalisant tout d'abord un diagramme de séquence système. Nous avons vu aux chapitres 1 et 2 l'intérêt de ce type de diagramme et les différents détails qui le concernent.



EXERCICE 5-2.

Diagramme de séquence système

Réalisez un diagramme de séquence système qui décrive le scénario nominal du cas d'utilisation TÉLÉPHONER.

solution

En se basant sur notre connaissance du domaine, nous allons décrire le scénario nominal de succès de communication entre un appelant et un appelé.

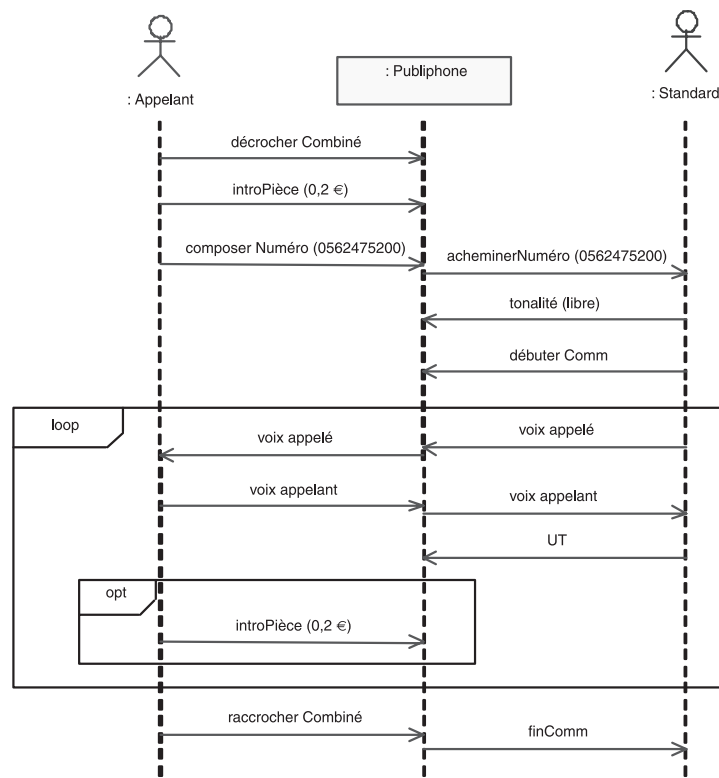
Comme cela est expliqué aux chapitres 1 et 2, nous utilisons la convention graphique suivante :

- l'acteur principal *Appelant* à gauche ;
- une ligne de vie représentant le *Publiphone* au milieu ;
- l'acteur secondaire *Standard* à droite.

Notez l'utilisation de deux fragments UML 2 :

- **loop** pour indiquer l'itération sur la phase de conversation avec le transport de la voix et le passage des unités de temps (UT) ;
- **opt** pour indiquer la possibilité d'ajouter des pièces pendant la conversation.

Figure 5-7.
Diagramme de séquence système du scénario nominal de Téléphoner



Nous n'avons pas représenté pour l'instant les réponses du Publiphone à l'appelant (en termes de tonalité par exemple) afin de ne pas alourdir ce premier schéma.



EXERCICE 5-3.

Diagramme de séquence système enrichi

Enrichissez le diagramme de séquence système précédent avec des activités internes intéressantes et quelques réponses du Publiphone à l'appelant.

En revanche, ne représentez plus la conversation afin de vous concentrer sur les « opérations système ».

Solution

Nous avons ajouté au diagramme précédent les activités internes importantes du Publiphone, telles que la vérification des pièces et la gestion du solde de l'appelant :

- incrémentation lors de l'introduction de pièces ;
- décrémentation lors du début de communication et à chaque UT.

Figure 5-8.

Diagramme de séquence système enrichi du scénario nominal de Téléphoner

