

Génie logiciel avec UML

Notions sur le langage UML adapté pour les cours du programme
Techniques de l'informatique

Table des matières

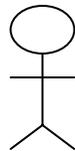
TABLE DES MATIÈRES.....	1
1. VUE FONCTIONNELLE DU SYSTÈME : CAS D'UTILISATION.....	3
1.1 LES ACTEURS	3
1.2 FRONTIÈRES DU SYSTÈME	3
1.3 LA RÉDACTION	4
1.3.1 <i>Les étapes de rédaction</i>	4
2. VUE STRUCTURELLE DU SYSTÈME : LE MODÈLE DE CLASSES	5
2.1 PERSPECTIVES DE REPRÉSENTATION DU DIAGRAMME DE CLASSES	5
2.1.1 <i>Perspective conceptuelle</i>	5
2.1.2 <i>Perspective spécifique</i>	5
2.1.3 <i>Perspective d'implantation</i>	5
2.2 LE MODÈLE CONCEPTUEL	6
2.2.1 <i>Création du modèle</i>	6
2.2.2 <i>Catégories de concepts</i>	6
2.2.3 <i>Catégories de relation</i>	7
2.2.4 <i>Attributs de concepts</i>	7
3. VUE COMPORTEMENTALE DU SYSTÈME : LE DIAGRAMME DE SÉQUENCE.....	8
3.1 RELATION AVEC LES CLASSES	8
3.2 SCHÉMATISATION.....	8
BIBLIOGRAPHIE	10

1. Vue fonctionnelle du système : cas d'utilisation

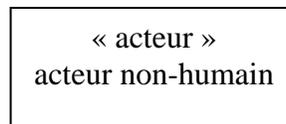
L'approche d'analyse logicielle basée sur les « besoins client » nécessite d'approfondir la connaissance des développeurs sur les interactions que veut avoir le client avec le futur système. L'approche des cas d'utilisation permet de mettre en lumière précisément la relation que le client désire avoir avec le système. Cette démarche permet de découvrir les acteurs et les frontières du système.

1.1 Les acteurs

Un acteur est une entité externe du système. Il peut être un humain ou un système informatisé. L'acteur consulte et/ou modifie l'état du système en en recevant ou en lui fournissant des données



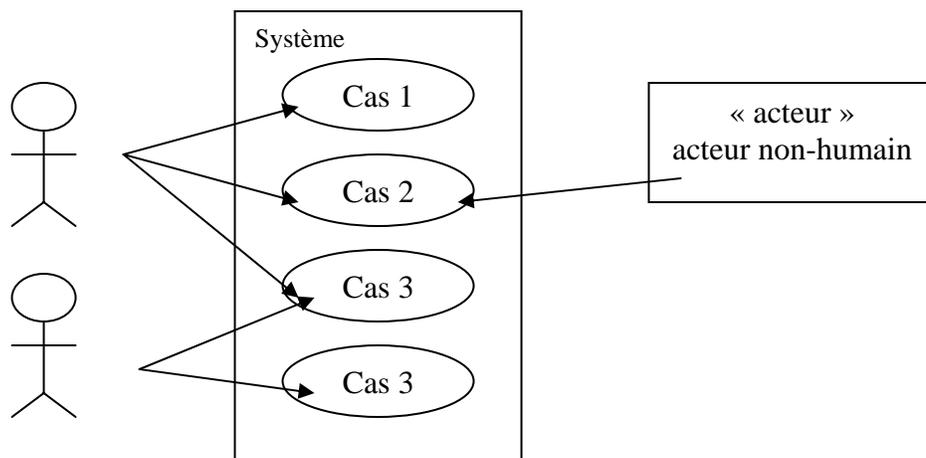
Acteur humain



Parmi ceux-ci, les utilisateurs client, les administrateur du système, les systèmes qui interagissent directement avec le système analysé.

1.2 Frontières du système

Le système devra posséder un comportement, soit effectuer des tâches, communiquer avec les acteurs et préserver son état. Le comportement particulier d'un système implique qu'il y est des frontières, soit que soient définies des tâches qui ne seront pas effectuées par le système. En émettant une liste des descriptions des cas d'utilisation, les frontières du système sont alors définies.



1.3 La rédaction

Une fois les frontières délimitées et le comportement représenté, les cas peuvent être spécialisés. Chaque cas est alors rédigé en détail. Puisque les cas font partie du domaine de la solution, la rédaction doit forcément être du vocabulaire du domaine. La séquence est communément un échange entre les actions des acteurs et les réponses du système.

- L'utilisateur clique sur l'affichage des enregistrements.
 - Le système affiche les enregistrements stockés.
- L'utilisateur sélectionne l'enregistrement désiré.
 - Le système ouvre la fenêtre des détails de l'enregistrement.

La rédaction détaillée mène dans un premier temps à une compréhension des interactions désirées et permet d'établir une entente avec le client quant au fonctionnement du système.

1.3.1 Les étapes de rédaction

- Extraire les acteurs.
- Cibler les actions par les verbes retrouvés dans le résumé de l'entrevue.
- Nommer les cas et écrire un résumé.
- Lier les acteurs aux cas.
- Détailler les cas.

Le niveau de détail des cas est relatif à la complexité de compréhension que représente ce cas. Plus le fonctionnement est particulier, plus le cas doit être détaillé. Inversement, les cas dont le déroulement est commun peuvent être décrits seulement par un résumé.

2. Vue structurelle du système : le modèle de classes

La découverte de la dynamique représentative des activités propres à un domaine d'affaire peut devenir facilement anarchique. La formalisation en vue de produire des classes est parfois instinctive et il est souhaitable d'encadrer cette activité. Ce document regroupe quelques notions permettant d'orienter la réflexion lors de la création de modèles conceptuels.

2.1 Perspectives de représentation du diagramme de classes

La conceptualisation du domaine du problème vise essentiellement à produire un modèle de classes et éventuellement programmer les classes. En chemin vers le modèle de classes, il est possible, en premier lieu, de modéliser sans tenir compte des contraintes de langage de programmation éventuellement utilisé. On dit alors que le niveau d'abstraction est élevé. Le chemin menant à un diagramme de classes peut être vu selon trois perspectives, soit conceptuelle, spécifique et d'implantation.

2.1.1 Perspective conceptuelle

Son niveau d'abstraction est élevé. On ne tient compte ici que du domaine du problème sans tenir compte des solutions possibles. Cette perspective ne permet que de comprendre les activités du domaine et d'acquérir le vocabulaire qui y est propre.

Le modèle conceptuel en résulte.

2.1.2 Perspective spécifique

Son niveau d'abstraction est moins élevé qu'au modèle conceptuel. On ajoute ici les propriétés et l'interface¹ de la classe. Le domaine de la solution est considéré dans cette perspective. Les contraintes de langage, le ou les environnements qui supporteront le système sont considérés.

Le diagramme de classes en résulte.

2.1.3 Perspective d'implantation

Le niveau d'abstraction est encore moins élevé ici. Le modèle de classes produit constitue une représentation très concrète des contraintes à considérer. Cette perspective est moins utilisée que les précédentes. Elle le sera dans les cas particuliers où la logique d'affaire à implanter est très peu commune. L'analyse selon cette perspective peut être utilisée sur quelques modules d'un système sans être étendue à tout le système.

Le diagramme de classes en résulte.

¹ Dans le langage orienté objet, l'interface d'une classe signifie l'ensemble de ses méthodes lui donnant un comportement qui lui est propre. L'interface est directement reliée aux responsabilités de la classe.

2.2 Le modèle conceptuel

2.2.1 Création du modèle

Les étapes naturelles pour arriver à la création d'un modèle conceptuel sont :

1. Faire une liste des concepts relatifs au domaine.
2. Les insérer dans le symbole du concept.
3. Établir des relations entre les concepts.
4. Ajouter des attributs qui augmentent le niveau d'information du modèle.

2.2.2 Catégories de concepts

Il existe des catégories de concepts communément rencontrées lors de modélisation de système.

Catégorie	Exemple
Physique ou tangible	Avion, atelier
Description	Description de vol d'avion, description de produit
Endroits	Magasin, aéroport, institution
Transactions	Vente, paiement, réservation
Item de transaction	Produit-quantité-prix
Rôle	Caissier, pilote, étudiant
Contenant	Magasin, wagon, entrepôt
Contenu	Item, passager, caisse
Système informatique externe	Autorisation de carte de crédit, contrôle de vol
Abstraction	Colère, agoraphobie
Organisation	Département, direction, administration
Évènement	Vente, arrivée, départ, écrasement, accident, rencontre
Processus ²	Vendre un produit, réserver une place, organiser une réunion.
Règles et politiques	Remboursement, annulation, crédit
Catalogue	Pièces, produits
Enregistrement et papiers légaux	Reçu, contrat, lègue
Manuel	Livre de réparation, microfiche.

² Un processus peut être considéré comme un suite d'évènements.

2.2.3 Catégories de relation

Catégories	Exemples
A est un partie de B	Roue-auto, facture-ligne facture
A est contenu dans B	Item-caisse, passager-véhicule, Description-catalogue
A est une description de B	Description-item, vol-description de vol
A est un ligne de facture de B	Ligne item- facture
A est enregistré en B	Vente-rapport, réservation-carnet de passagers
A est un membre de B	Caissier-magasin, pilote-ligne aérienne
A est un sous-unité de B	Département-cégep, vente-industrie
A utilise B	Pilote-avion, Caissier-caisse
A communique avec B	Client-vendeur, étudiant-prof
A est relatif à une transaction de B	Client-paiement, passager-billet
A est une transaction relative à une autre transaction B	Paiement-vente, réservation-annulation
A est voisin de B	Ville-ville, local-local
A est la propriété de B	Caisse-magasin, avion-ligne aérienne

2.2.4 Attributs de concepts

Les attributs doivent être le plus simple possible. La plupart du temps ils représenteront des valeurs de données simples : booléen, date, nombre, chaîne de caractères, temps.

Des valeurs plus complexes peuvent être incluses comme attributs : adresse, couleur, numéros en tout genre, code postal.

3. Vue comportementale du système : le diagramme de séquence

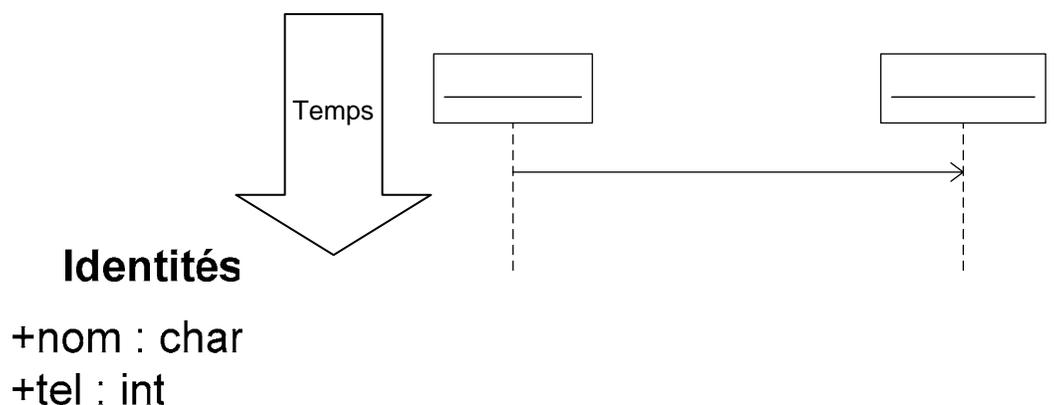
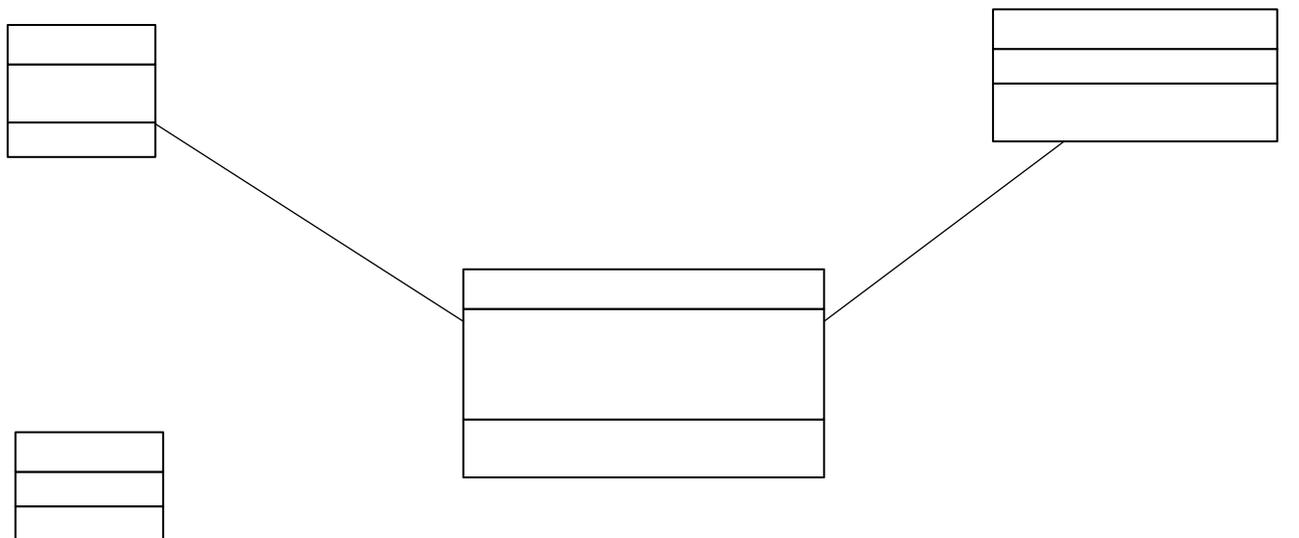
Le diagramme de séquence met en action les classes intervenant dans le scénario d'un cas d'utilisation.

3.1 Relation avec les classes

La cohérence entre les cas d'utilisation et les diagrammes de classes se trouve dans les diagrammes de séquence. Ce type de schéma donne une représentation de la communication entre les différentes classes mises en action dans le déroulement d'un cas. L'information du diagramme met en particulièrement les messages qui sont envoyés entre les classes (fonctions et retour) et l'ordonnancement dans le temps de ces envois de messages.

3.2 Schématisation

Un diagramme de séquence typique ressemblerait à ceci :



Le diagramme peut prendre des formes plus complexes selon les échanges. Cependant, peu importe la complexité, il y a toujours un appel de fonction entre deux classes et un retour de fonction exprimé par une réponse dans le diagramme.

Bibliographie

LARMAN, Craig; Applying UML and patterns; Prentice-Hall; 1998; 507 pages.

FOWLER, Martin; UML Distilled Second Édition; Addison-Wesley; 2000; 186 pages.

ROQUES, Pascal; UML2 par la pratique; Eyrolles; 2006; 357 pages.