

## Conseils supplémentaires – Modèle du domaine

Ces informations complètent le contenu des textes 2.6 **Modèles du domaine : visualisation des concepts** (chapitre 9, pp. 143-180) et 2.7 **Affinement du Modèle de Domaine** (chapitre 26, pp. 491-504) ou mettent en évidence certaines idées présentes dans ces textes.

### Définitions

#### *Modèle du domaine*

- C'est une représentation visuelle des classes conceptuelles ou des objets du monde réel dans un domaine donné.
- Il est aussi nommé modèle conceptuel, modèle objet du domaine, modèle objet d'analyse ou encore diagramme de classes conceptuelles.
- Il est constitué de classes conceptuelles, d'associations entre classes conceptuelles et d'attributs qui décrivent chacune des classes conceptuelles.

#### *Attributs*

- Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeurs possibles partagées par tous les objets de la classe. À tout instant, chaque objet d'une classe porte une valeur spécifique pour chaque attribut de la classe.

#### *Association*

- Une association est une relation entre des classes (ou plus précisément des instances de classes) qui indique une connexion significative ou intéressante.
- Une association représente une relation sémantique entre deux classes; elle est représentée par une ligne entre deux classes; elle peut avoir un nom, un rôle pour chacune des classes impliquées dans l'association, une cardinalité et une direction.
- Il existe plusieurs types de relation :
  - Agrégation : l'agrégation est un type d'association utilisé pour modéliser des relations de tout à partie entre différentes entités. Le « tout » est appelé composite. C'est un type spécial d'association utilisé pour indiquer que des

objets participants sont assemblés et configurés ensemble pour créer un nouvel objet plus complexe.

- Composition : la composition ou l'agrégation composite signifie que le composant est membre d'un seul objet composite et qu'il en est dépendant pour sa création et sa destruction. La composition est utilisée pour des agrégations où l'étendue de vie des parties dépend de celle du tout. Le tout a le contrôle sur la création et la destruction des parties; une partie ne peut être membre que d'un seul objet composite.
- Généralisation (ou héritage, spécification) : la généralisation est l'activité qui consiste à identifier les éléments communs aux différentes classes conceptuelles et à définir des relations de superclasses (concept général) et de sous-classes (concept particulier). La définition d'une classe conceptuelle est plus générale ou plus englobante que celle d'une sous-classe.

#### *Classe d'association*

- C'est un mécanisme pour ajouter des propriétés à l'association elle-même.

#### *Quel rôle joue le modèle du domaine ?*

- Identifier et structurer l'information importante du domaine pour le problème à traiter.
- Le modèle répond à la question « Qu'est-ce qu'il y a de significatif dans le monde réel ou la pratique permettant de traiter le problème que le système devra résoudre? ».
- C'est une vue statique sur le domaine.

#### **Quelques pièges à éviter**

- Les modèles du domaine ne sont pas des modèles de composants logiciels : ni les artefacts logiciels (comme les fenêtres graphiques ou les bases de données), ni les responsabilités ou les méthodes ne doivent y figurer.
- Une erreur courante lors de la création d'un modèle du domaine consiste à représenter quelque chose comme un attribut alors que ce devrait être un concept. N'utilisez pas un concept comme attribut d'un autre concept. À la place, il faut modéliser le concept comme une classe et le relier à la classe de l'attribut par une association
- Vous ne devez pas utiliser d'attributs pour lier des classes conceptuelles dans un modèle du domaine.
- Les associations ne sont pas des déclarations relatives aux flux de données, aux variables d'instance, ni aux connexions entre objets de solution logiciel.
- Ne pas confondre objet physique et objet logique, en d'autres termes une entité et sa description.
- Ne pas confondre relation de composition et relation d'agrégation.
- Ne pas abuser de la généralisation.

- Une classe ne doit pas être décrite en termes de ce qu'elle « FAIT », mais de ce qu'elle « EST ».
- Si les classes correspondent à une division fonctionnelle du problème, il y a un problème.
- Si vous ne pouvez pas identifier des instances de la classe, alors il doit y avoir un problème.
- Si la classe a seulement une instance, il y a peut-être un problème, car il est rare qu'une classe n'ait qu'une instance.

## **Conseils**

### *Sur le modèle du domaine*

- Un modèle du domaine n'est pas absolument juste ou faux. C'est un outil de communication.
- Un bon modèle du domaine capture les abstractions essentielles et les informations requises pour comprendre le domaine dans le contexte des besoins effectifs, avec ses concepts, sa terminologie et ses relations.
- Mieux vaut sur-spécifier un modèle du domaine avec de nombreuses classes à granularité fine que le sous-spécifier.
- Les premières classes candidates identifiées dans cette phase doivent être des concepts connus des utilisateurs du système, ce qu'on appelle couramment les objets métier.
- Donnez-leur un nom significatif.
- La recherche de classes conceptuelles est plus importante que celle des associations.

### *Pour utiliser une classe d'association*

- Quand un attribut est lié à une association.
- Quand il y a une association \* -- \* entre deux classes conceptuelles.

### *Pour l'utilisation de la relation d'agrégation*

- La durée de vie du composant est limitée à celle du composite : la partie dépend du tout.
- Il existe un assemblage intégral logique ou physique évident.
- Certaines propriétés du composite, comme la localisation, s'étendent aux composantes.

### *Pour l'utilisation de la relation de généralisation*

- Raison pour spécialiser : la sous-classe a des attributs ou des associations supplémentaires pertinents ou un comportement différent, ou on veut agir sur elle différemment.

- Raison pour généraliser et créer une super-classe : les classes représentent des variantes d'un concept similaire ou on peut factoriser des attributs des classes.

*En vrac*

- Identifiez les attributs de ces objets correspondant à une nécessité de mémoriser une information que vous avez repérée lors de l'analyse des besoins.
- Reliez des classes conceptuelles par des associations, non par des attributs.
- Associez aux classes, sous forme de commentaires, d'autres informations du domaine les concernant.
- Concentrez-vous sur les associations pour lesquelles il est nécessaire de mémoriser la relation pendant un certain temps.
- Soyez très précis avec les multiplicités des associations. Elles doivent être vérifiées à tous les moments du cycle de vie des instances.
- Gardez uniquement comme attributs les propriétés simples des classes que le système doit mémoriser et manipuler.
- Concernant les associations d'agrégation et de composition : dans le doute, abstenez-vous!

## **Références**

Roque, P., Vallée, F. 2000. *UML en action*. Éditions Eyrolles.

Villalobos, J. Notes du cours *Génie logiciel orienté objets*. Université de Los Andes, Bogota, Colombie.

Fowler, M. 2000. *UML*. Éditions Campus Press.

Pender, T. 2002. *UML Weekend Crash Course*. Wiley Publishing, Inc.