Glossaire UML

Abstraction

Démarche qui consiste à ne considérer que certains éléments d'un problème, pour des raisons de pertinence et/ou d'indépendance.

Acteur ("actor")

Interlocuteur d'un système. Le système ne connaît du domaine que les acteurs: utilisateurs, équipements, autres systèmes. Ne pas confondre l'acteur (représenté dans les cas d'usages et interactions) et l'objet du système auquel il est associé (représenté par un stéréotype dans le modèle structurel).

Action

Intervention d'un objet affectant l'état du système. (cf. Activité)

Activité

Action dont le déroulement est susceptible d'être affecté par des événements définis en dehors de l'objet (composant ou système) qui la réalise. Selon le niveau d'abstraction une action peut correspondre à un scénario comme à une opération.

Agent

Objet ayant la capacité de créer une interaction. On parle aussi d'objet actif lorsqu'il s'agit d'une activité (flot d'exécution).

Agrégation ("agrégation")

Cas particulier d'association qui ajoute l'idée d'appartenance. (cf. Composition)

Association("association")

Description des liens pouvant exister entre les objets de deux classes (ou plus). Synonyme de relation.

Association qualifiée ("qualified")

Association qui permet un accès sélectif aux objets associés en fonction de la valeur du critère de qualification.

Analyse

Spécification des besoins auxquels doit répondre un système (ou un composant). L'analyse traite de l'interface et du comportement sans se préoccuper de l'implémentation. (cf. Conception)

Architecture

Spécifie la structure d'un système. L'architecture fonctionnelle porte sur les services communs, et les règles qui permettent aux éléments de collaborer. L'architecture technique traite des moyens mis en œuvre. On parle aussi d'architecture applicative

pour décrire le découpage en sous-systèmes (découpage vertical) et couches (découpage horizontal).

Atomicité

Caractéristique d'une action qui, dans un domaine donné, doit être exécutée totalement où pas du tout. Formellement cela signifie que l'action est instantanée pour le domaine considéré, ou encore qu'aucun événement significatif ne peut intervenir durant son exécution, ou bien encore que le début et la fin forment un seul et unique événement.

Attribut ("attribute")

Propriété d'un type d'objet. Un attribut n'a pas d'identité et possède une valeur distincte pour chacun des objets.

Attribut de classe ("class attribute")

Attribut dont la valeur est commune à tous les objets du type. Ne pas confondre avec une constante ou une valeur par défaut.

Cas d'usage ("use case")

Ensemble d'activités concourant à la réalisation d'un résultat identifiable, en réponse à l'intervention d'un acteur. (cf. Processus opérationnel, Scénario)

Classe ("class")

Description d'un ensemble d'objets. Au sens strict une classe décrit (totalement ou partiellement) une implémentation. On parlera de type pour une description limitée à l'interface (attributs et opérations), indépendante de toute implémentation. (cf. Type)

Classe abstraite ("abstract class")

Description générique d'un ensemble d'objets. En termes d'implémentation une classe abstraite factorise les éléments communs à plusieurs sous-classes. Elle est par définition incomplète. (cf. Classe concrète)

Classe concrète ("concrete class")

Description opérationnelle d'un ensemble d'objets. En termes d'implémentation une classe concrète fournit les éléments nécessaires à l'instanciation des objets. Elle est par définition complète. (cf. Classe abstraite)

Classe active ("active class")

Classe dont les instances peuvent être à l'origine d'un flot d'exécution. (cf. Flot d'exécution)

Classe Associative ("associative class")

Classe d'objets dont l'existence (et donc l'identité) dépend de l'existence d'un lien entre deux autres objets (ou plus).

Classe paramétrée ("parametrized class")

Classe générique construite à partir de méthodes génériques ("Template"). Une classe paramétrée est une implémentation de méta classe. Les classes sont obtenues en spécifiant le type du paramètre utilisé par les méthodes. (cf. Template)

Collaboration ("collaboration")

Ensemble de rôles et de mécanismes assurant la réalisation d'un scénario.

Composition ("composition")

Cas particulier d'agrégation qui ajoute la dépendance existentielle du composant par rapport au composé. (cf. Agrégation)

Composant ("component")

Élément physique et interchangeable d'un système qui réalise un ensemble d'interfaces.

Conception

Spécification de l'implémentation d'un système (ou d'un composant). La conception prend comme point de départ l'interface et le comportement requis. (cf. Analyse)

Conteneur

Objet fonctionnel crée par le système pour gérer des ensembles d'objets (Équivalent à collection).

Contrainte

Expression qui limite l'existence et/ou la valorisation des objets, ou l'occurrence et/ou les modalités des événements et activités.

Contrôleur

Objet fonctionnel crée par le système pour implémenter les mécanismes d'une collaboration.

Couche ("layer")

Regroupement de composants assurant un ensemble de services partagés par plusieurs sous-systèmes clients. (cf. Sous-système, Architecture).

Couplage

Mesure de la dépendance liée à l'envoi d'un message. L'objet émetteur doit connaître la signature du message. Toute modification affectera la description de l'objet émetteur. L'objectif est de minimiser cette dépendance. (cf. Dépendance)

Délégation

Mécanisme par lequel un objet ne traite pas lui même (ou traite partiellement) un message, et le redirige vers un autre objet.

Dépendance ("dependancy")

Lien entre deux éléments dont la description de l'un est subordonnée à la description de l'autre. (cf. Couplage)

Dépendance d'exécution

Lien qui conditionne l'activité d'un objet à l'activité d'un autre dans le contexte d'un collaboration. L'objectif est de minimiser cette dépendance. (cf. Couplage)

Déploiement

Mise en œuvre des composants dans leur environnement opérationnel.

Dérivé ("derived")

Un élément dérivé est un élément (attribut, lien, objet) qui peut à tout instant être reproduit (calculé ou construit) à partir des objets du système sans que cela affecte l'état du système.

Descriptif

Type dont les instances correspondent à des sous-ensemble d'un autre type. Ces sous ensemble ne sont pas nécessairement des sous-types. (cf. Powertype)

Diagramme d'activité ("activity diagram")

Cas particulier d'un diagramme d'état dans lequel les états représentent des activités. Lorsque la modélisation se fait sur plusieurs niveaux d'abstraction, les transitions d'un diagramme d'état peuvent correspondre à des activités. (cf. Diagramme d'état)

Diagramme d'états ("state diagram")

Représentation d'un comportement en termes d'états et de transitions provoquées par des événements. Un diagramme d'état fournit une spécification exécutable (un automate) et correspond donc à une implémentation. (cf. État, Transition, StateCharts)

Diagramme de cas d'usage ("use case diagram")

Représentation des cas d'usage. Cet outil est utilisé pour gérer les fonctionnalités du système, et plus généralement les spécifications externes d'un sous-système ou d'un composant.

Diagramme de classes ("class diagram")

Représentation statique des types ou classes et de leurs relations. cet outil est utilisé aussi bien pour les modèles d'analyse que de conception. Dans une approche traditionnelle on se limite à des types de données sans y associer d'opérations.

Diagramme de collaboration ("collaboration diagram")

Représentation des collaborations à partir d'un diagramme d'objets. Les collaborations peuvent être représentées plus précisément à l'aide d'un diagramme de séquence. (cf. Diagramme de séquence, Diagramme d'objets)

Diagramme de composants ("component diagram")

Représentation de l'organisation et des dépendances des composants

Diagramme de déploiement("deployment diagram")

Représentation de l'architecture technique du système: Composants et nœuds. (cf. Composant, Nœud)

Diagrammes d'interactions ("interaction diagram")

Au sens étroit regroupe les diagrammes de collaboration et de séquence. Au sens large on ajoute le diagramme d'activités.

Diagramme d'objets ("object diagram")

Représentation des objets participant à un scénario, ainsi que de leurs liens (visibilité statique).

Diagramme de séquence ("séquence diagram")

Représentation détaillée des collaborations en termes d'objets, de messages et d'activités.

Domaine

Contexte applicatif caractérisé par une communauté d'objectifs et de ressources. Plus concrètement les acteurs, objets et événements y sont identifiés de la même manière.

Encapsulation

Mécanisme qui assure l'étanchéité de l'implémentation: Une opération est définie indépendamment des méthodes qui l'implémentent.

Entité ("entity ")

Représentation persistante dans le système d'un objet de gestion. Une entité possède une identité qui la distingue des autres. Par ailleurs elle possède au moins un attribut dont la valeur est pertinente pour le domaine concerné. (cf. Objet de gestion)

Énumération ("énumération")

Stéréotype qui désigne des objets dont la fonction est de gérer la référence à d'autres objets.

État ("state ")

Représente la situation d'une activité ou d'un objet. Les états (de l'une ou de l'autre) doivent significatifs par rapport à un comportement ou une collaboration. Leur nombre doit être fini. (cf. Diagramme d'état, Transition)

Événement ("event")

Intervention d'un acteur ou modification de l'état d'un objet. Un événement est instantané et significatif dans un domaine donné.

Fil de contrôle ("thread")

Équivalent d'un processus dépourvu de ressources propres. Correspond à l'exécution d'un flot de contrôle à l'intérieur d'une collaboration. (cf. Flot de contrôle, Processus).

Flot de contrôle ("flow of control")

Ensemble des activités générées par un événement. Synonyme de flot d'exécution. (cf. Fil de contrôle, Processus).

Flux de données ("data flow")

Échange d'informations non accompagné par un changement d'état, donc sans événement associé.

Flux de contrôle ("control flow")

Échange d'informations accompagné par un changement d'état, donc avec un événement associé. (cf. Message)

Framework

Cas particulier de pattern traitant d'un problème de collaboration. Contrairement à une méta classe un framework peut être directement implémenté, mais à la différence d'un pattern classique son implémentation est distribuée parmi les objets souhaitant participer à la collaboration en question. (cf. Pattern)

Généralisation

Description simplifiée mais complète d'un ensemble d'objets ou d'actions. Cette description est complète (suffisante pour créer des objets ou réaliser des scénarios) mais elle ne couvre pas toutes les variantes. A comparer avec l'abstraction, qui fournit une description incomplète mais qui couvre formellement toutes les variantes. (cf. Spécialisation, Abstraction)

Héritage

Mécanisme par lequel les spécifications d'une classe (attributs, opérations, ou méthodes) s'appliquent aux sous-classes.

Identité

Propriété d'un objet qui le distingue des autres indépendamment de l'état dans lequel il se trouve.

Instance

Réalisation d'une classe. Synonyme d'occurrence. La notion d'objet est plus large et inclut aussi bien les classes elles mêmes que les objets externes au système. (cf. Objet)

Instanciation

Mécanisme par lequel une classe crée une instance.

Intégrité

Cohérence des objets persistants en regard des invariants. L'intégrité référentielle porte sur les conditions d'existence des objets, l'intégrité fonctionnelle concerne l'état des objets.

Interface ("boundary")

Un des trois stéréotypes de base (avec les entités et les contrôleurs) qui dans la pratique ont été intégrés à UML. Ce stéréotype représente les objets transitoires qui réalisent les échanges entre le système et les acteurs. Il est généralement associé à la vue logique des flux d'information. Ne pas confondre avec l'interface des classes ("interface").

Interface ("interface")

Spécifications externes d'un ensemble d'opérations fonctionnellement homogènes. La notion d'interface est plus restreinte que la notion de type: Une interface se limite aux opérations, alors qu'un type peut avoir des attributs. Elle renvoie à la notion de service, alors que le type est utilisé pour décrire des objets aussi bien que des comportements. Par ailleurs une interface peut être implémentée en tant que telle. On obtient un objet physique qui matérialise les conditions d'une collaboration. Ne pas confondre avec l'interface des systèmes ("boundary").

Invariant

Condition qui doit être maintenue au niveau du domaine. (cf. Variante)

Lien ("link ")

Instance d'une relation.

Message ("message ")

Mécanisme par lequel un objet communique avec un autre. Un message est supposé provoquer l'exécution d'une opération par l'objet destinataire. Il faut distinguer: Le message et l'opération: un même message peut déclencher des opérations différentes selon l'objet que le reçoit. (cf. Polymorphisme). Le message et la réponse: un événement est associé à la réception d'un message. Si la réponse est instantanée il n'y a pas de message associé à la réponse.

Meta classe ("metaclass")

Classe dont les instances sont elles mêmes des classes. (cf. Descriptif)

Méthode ("method ")

Implémentation d'une opération. (cf. Encapsulation)

Navigation

Accessibilité des objets exprimée en termes de relations et de conditions.

Nœud ("node")

Ressource physique supportant l'exécution, la persistance ou la transmission des composants.

Objet

Élément identifiable caractérisé par les états qu'il peut prendre et les opérations qu'il peut réaliser. En termes d'implémentation la notion d'objet couvre l'ensemble des éléments physiques identifiables par le système, ce qui peut inclure les classes. (cf. Instance)

Objet de gestion

Objet identifié et géré par le domaine. Les objets de gestion sont représentés de manière persistante dans le système par les entités. (cf. Entité, Objet fonctionnel)

Objet de conception

Objet défini pour répondre aux besoins du concepteur. Plus restrictif que la notion d'objet fonctionnel.

Objet fonctionnel

Objet identifié et géré par le système, indépendamment du domaine. Un objet fonctionnel ne représente rien et ne se justifie que par ce qu'il fait. Il peut être persistant mais sa persistance n'est pas significative en dehors du système. (cf. Objet de gestion)

Objet métier

Implémentation (accès partagé, persistance, intégrité) d'un objet de gestion dans le contexte d'une architecture distribuée. (cf. Objet de gestion)

Opération ("opération")

Service supporté par un type d'objet et implémenté par les classes. Une opération est identifiée par sa signature: son nom et le type de ses arguments. Les opérations sont implémentées par des méthodes (définies par les classes) et exécutées dans le contexte des objets.

Opération de classe ("class opération")

Opération exécutée dans le contexte de la classe et non pas des objets.

Pattern

Dans son interprétation restreinte un pattern correspond à une solution de référence pour un problème standard. L'ouvrage d'Eric Gamma et de ses collègues (la "bande des quatre") a fournit un cadre largement accepté pour identifier ces problèmes.

Habituellement un pattern est suffisamment précis pour être directement implémenté. (cf. Framework)

Paquetage ("package")

Regroupement d'éléments logiques utilisés en cours de développement. Les paquetages peuvent correspondre à un découpage opérationnel (cf. Sous-systèmes) ou ne répondre qu'aux besoins du processus de développement. Ils fournissent un espace d'adressage privé et peuvent être hiérarchisés.

Polymorphisme

On parle de polymorphisme statique lorsqu'un même message peut prendre plusieurs formes, c'est à dire plusieurs signatures. Dans une approche objet ce type de polymorphisme n'apporte rien de nouveau et relève de l'implémentation. On parle de polymorphisme (dynamique) lorsque la sémantique de l'opération est déterminée à l'exécution, en fait à la réception du message. C'est alors, et seulement alors que l'on connaît l'objet, donc la classe, donc la méthode. En d'autres termes on peut avec un même message (même nom et même forme,) avec différentes interprétations.

Post-condition ("post condition")

Condition supposée réalisée lorsqu'une action s'est déroulée normalement. Le respect de ces condition relève de l'objet qui implémente l'action. (cf. Pré-condition)

Powertype ("powertype")

Type dont les instances correspondent à des sous-ensemble d'un autre type. Ces sous ensemble ne sont pas nécessairement des sous-types. (cf. Descriptif, Méta classe)

Pré-condition ("pre condition")

Condition supposée réalisée lors du déclenchement d'une action. Le respect de ces conditions relève de l'objet qui demande l'action. (cf. Post-condition)

Processus ("process")

Ensemble des activités générées par un événement externe et capable de s'exécuter indépendamment, c'est à dire sur ses seules ressources. Correspond à l'exécution d'une collaboration. (cf. Flot de contrôle, Fil de contrôle).

Processus opérationnel ("business process")

Dans le contexte de l'ingénierie des processus il s'agit d'un flux d'activités concourant à la réalisation d'un besoin identifié par le domaine. Les processus opérationnels sont l'équivalent au niveau organisationnel des cas d'usage. (cf. Cas d'usage)

Rôle

Pour une relation: nom du type de la cible dans le contexte d'origine. Pour un cas d'usage: description de la place tenue par un objet dans la réalisation du cas d'usage. Par ailleurs nous avons introduit un stéréotype (représenté par un masque) qui combine ces deux aspects: le stéréotype signale une contrainte d'identité entre un objet les rôles qu'il tient (ses avatars) dans les cas d'usage.

Relation ("relationship")

Synonyme d'Association.

Scénario ("scénario")

Séquence d'actions qui réalise un cas d'usage et correspond donc à une variante.

Sous-système

Regroupement de composants assurant un ensemble de services fonctionnellement homogènes. Parfois utilisé pour désigner un ensemble intégré de services nécessaires à la réalisation d'un (ou plusieurs) cas d'usage. Les sous-systèmes sont gérés par des paquetages stéréotypés. (cf. Paquetage, Couche)

Spécialisation

Inverse de la généralisation. Permet de préciser une description pour couvrir des situations spécifiques. La spécialisation peut se faire par extension (ajout de nouvelles caractéristiques ou opérations) ou par restriction (définition d'un sous-ensemble). La spécialisation doit toujours respecter le principe de substitution. (cf. Généralisation, Substitution)

StateCharts

Langage de modélisation permettant de représenter formellement des automates d'états finis sur plusieurs niveaux d'abstraction. Repris dans UML. (cf. Diagramme d'états).

Stéréotype ("stereotype")

Extension des éléments de modélisation définis par UML. Un stéréotype précise la sémantique d'un élément, et donc la manière dont il doit être utilisé dans un modèle.

Signature ("signature")

La signature d'une opération regroupe son nom et le type de ses arguments.

Substitution (Principe de Liskov)

Les opérations applicables à une classe doivent s'appliquer à toutes ses sous-classes. Dans sa version forte toute opération définie sur une classe doit être applicable avec la même sémantique à toutes les sous-classes. En d'autres termes l'héritage ne peut modifier que l'implémentation des opérations.

Template ("template")

Mécanisme qui permet de définir une méthode sans avoir à définir le type de ses arguments. Uniquement implémenté par le langage C++.

Type ("type")

Spécification d'un ensemble d'attributs et d'opérations. Un type peut être implémenté par une classe, ou par d'autres mécanismes. (cf. Classe)

Transition ("transition")

Passage d'un état à un autre provoqué par un événement. Une transition peut être conditionnée par une garde et provoquer une action. (cf. État)

Variante

Condition qui détermine le déroulement d'un cas d'usage. Par extension, scénario qui réalise les actions associées à la variante. (cf. Invariant, Scénario)