Glossaire du module CSC4102

Denis Conan

Janvier 2019

Avertissement :
La plupart des définitions sont fournies en anglais et nous n’indiquons pas l’origine des définitions. Ce travail reste à faire.
A

acteur/actor [class]
A predefined stereotype of class denoting an entity outside the system that interacts with use cases.

action
A computational or algorithmic procedure.

agrégation/aggregation
A special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part association class.

analyse/analysis
The part of the software development process whose primary purpose is to formulate a model of the problem domain. Analysis focuses what to do, design focuses on how to do it.

artefact
A piece of information that is used or produced by a software development process. An artefact can be a model, a description or software.

association
A relationship that describes a set of links.

association binaire/binary association
An association between two classes.

association n-aire/n-ary association
An association among three or more classes. Each instance of the association is an n-tuple of values from the respective classes.

attribut/attribute
A named property of a class.

C

cardinalité/cardinality
The number of elements in a set.

cas d’utilisation/use case [class]
A class that defines a set of use case instances.

classe/class
Une classe décrit une collection d’objets ayant mêmes attributs et mêmes méthodes ; étant donnée une classe, on peut créer des objets de la classe, on dit aussi des instances de la classe. Chaque instance a sa propre identité et les identificateurs ne sont pas accessibles aux utilisateurs. En d’autres termes, une classe définit un type et toutes les instances de la classe sont de ce type. Ou encore, une classe représente la mise en œuvre d’un type (abstrait).
classe abstraite/abstract class
A class that cannot be directly instantiated. 2

classe concrète/concrete class
A class that can be directly instantiated. 2

classe d’association/association class
A modeling element that has both association and class properties. An association class can be seen as an association that also has class properties, or as a class that also has association properties. 2

composant/component
An executable software module with identity and a well-defined interface. 2

composition
A form of aggregation with strong ownership and coincident lifetime as part of the whole. Parts with non-fixed multiplicity may be created after the composite itself, but once created they live and die with it (i.e., they share lifetimes). Such parts can also be explicitly removed before the death of the composite abstract class. 2

conception/design
The part of the software development process whose primary purpose is to decide how the system will be implemented. During design strategic and tactical decisions are made to meet the required functional and quality requirements of a system. 2

D
dépendance/dependency
A relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element). 2

diagramme/diagram
A graphical presentation of a collection of model elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements). 2

E
élément de modèle/model element
An element that is an abstraction drawn from the system being modeled. 2

élément dérivé/derived element
A model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information. 2

encapsulation
A means to bind together code and the data it uses. 2

espace de nommage/namespace
A part of the model in which the names may be defined and used. Within a namespace, each name has a unique meaning. 2

état/state
A condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event. 2
état composite/composite state
A state that consists of substates. 2

événement/event
A significant occurrence. An event has a location in time and space and may have parameters. In the context of state diagrams, an event is an occurrence that can trigger a state transition. 2

exigence/requirement
A desired feature, property, or behavior of a system. 2

G

généralisation spécialisation/generalisation
A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed. 2

généralisation spécialisation multiple/multiple inheritance
A semantic variation of generalization in which a type may have more than one supertype. 2

H

héritage/inheritance
L’héritage définit une relation entre classes, les sous-classes héritant de la structure et du comportement des « super-classes ». De plus, une sous-classe peut redéfinir des méthodes de ses « super-classes ». On parlera de spécialisation dans la mesure où on ajoute des attributs ou des méthodes particulières à une classe pour réaliser une sous-classe, et de généralisation d’un ensemble de sous-classes dans la mesure où on met en commun des attributs et des méthodes pour constituer une super-classe. 2

I

interaction
A behavioral specification that comprises a set of message exchanges among a set of objects within a particular context to accomplish a specific purpose. An interaction may be illustrated by one or more scenarios. 2

interface
The use of a type to describe the externally visible behavior of a class, object, or other entity. In the case of a class or object, the interface includes the signatures of the operations. 2

L

ligne de vie/object lifeline
A line in a sequence diagram that represents the existence of an object over a period of time. 2

M

machine à états/state machine
A behavior that specifies the sequences of states that an object or an interaction goes through during its life in response to events, together with its responses and actions. 2
message
A communication between objects that conveys information with the expectation that activity will ensue. The receipt of a message is normally considered an event.  

message asynchrone/asynchronous message
A message where the sending object does not pause to wait for results.  

message synchrone/synchronous message
A message where the sending object pauses to wait for results.  

méta-modèle/metamodel
A model that defines the language for expressing a model.  

méthode/method
The implementation of an operation. The algorithm or procedure that effects the results of an operation.  

modèle/model
A semantically closed abstraction of a system.  

multiplicité/multiplicity
A specification of the range of allowable cardinalities that a set may assume. Multiplicity specifications may be given for roles within associations, parts within composites, repetitions, and other purposes. Essentially a multiplicity is a (possibly infinite) subset of the non-negative integers.  

N
navigabilité/navigability
The ability for objects of a class at one end of an association to retrieve objects from the other end.  

O
objet/object
An entity with a well-defined boundary and identity that encapsulates state and behavior. State is represented by attributes and relationships, behavior is represented by operations and methods. An object is an instance of a class.  

opération/operation
A service that can be requested from an object to effect behavior. An operation has a signature, which may restrict the actual parameters that are possible.  

P
paquetage/package
A general purpose mechanism for organizing elements into groups. Packages may be nested within other packages. A system may be thought of as a single high-level package, with everything else in the system contained in it.  

polymorphisme ad hoc/ad hoc polymorphism
C’est le polymorphisme de la surcharge.  

d polymorphisme d inclusion/inclusion polymorphism
C’est le polymorphisme de la redéfinition. Il est aussi appelé polymorphisme de sous-type.
polymorphisme de sous-type/subtype polymorphism
Synonyme de polymorphisme d’inclusion.

polymorphisme paramétrique/parametric polymorphism
C’est le polymorphisme des types paramétrés (generics).

postcondition/postcondition
The postconditions for an Operation define conditions that will be true when the invocation of the Operation completes successfully, assuming the preconditions were satisfied. These postconditions shall be satisfied by any implementation of the operation.

précondition/precondition
The preconditions for an Operation define conditions that shall be true when the Operation is invoked. These preconditions may be assumed by an implementation of this Operation. The behavior of an invocation of an Operation when a precondition is not satisfied is not defined in UML.

processus de développement/development process
A set of partially ordered steps performed for a given purpose during software development, such as constructing models or implementing models.

processus de modélisation/modelling time
Refers to something that occurs during a modeling phase of the software development process. It includes analysis time and design time. Usage note: When discussing object systems it is often important to distinguish between modeling-time and run-time concerns.

processus métier/business process
Est un ensemble coordonné de plusieurs activités, mobilisant plusieurs acteurs, en vue de satisfaire une demande d’un ou plusieurs biens, ou d’un ou plusieurs services.

R
redéfinition/overriding
La redéfinition est une construction du langage qui permet à une classe enfant de fournir une version spécialisée d’une opération, qui existe avec le même prototype dans la classe parente. La définition de l’opération de la classe enfant remplace celle de la classe parente. Selon les langages qui fournissent ce mécanisme, le choix de l’opération à exécuter peut être effectué à la compilation (liaison dite statique) ou à l’exécution (liaison dite dynamique ou tardive [en anglais, late binding]). La redéfinition est aussi appelée « polymorphisme d’inclusion »: le même nom d’opération possède « plusieurs formes » selon le type courant par opposition au polymorphisme dit ad hoc utilisé lors d’une surcharge ou au polymorphisme dit paramétrique utilisé dans les classes paramétrées (par un type).

règle de gestion/business rule
Encadre, limite, définit les conditions de réalisation d’une fonctionnalité. Ce peut être une règle de calcul, de confidentialité, de cohérence, d’intégrité, etc.

rôle/role
The named specific behavior of an entity participating in a particular context. A role may be static (e.g., an association role) or dynamic (e.g., a collaboration role).
sous-classage/subclassing

Subclassing is where one type is replaced by another, which also systematically replaces the original functions with new ones appropriate to the new type. It is a more complex concept than subtyping since it involves the polymorphic approach.

sous-typage/subtyping

Subtyping is where an object of one type may safely be substituted where another type was expected. This involves no more than coercing the supplied subtype object to a supertype and executing the supertype's functions: no dynamic binding is implied or required. The coerced object then behaves in exactly the same way as expected.

stéréotype/stereotype

A new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes. Certain stereotypes are predefined in the UML, others may be user defined.

surcharge/overloading

La surcharge est une construction du langage qui permet de disposer de différentes versions d'une même opération selon le nombre et le type des arguments fournis: par exemple, void afficher(Voiture v) et void afficher(Agenda a), l'affichage étant différent pour les types Voiture et Agenda. Selon les langages qui fournissent ce mécanisme, le choix de l'opération à exécuter peut être effectué à la compilation (liaison dite statique) ou à l'exécution (liaison dite dynamique ou tardive en anglais, late binding). La surcharge est aussi appelée « polymorphisme ad hoc »: le même nom d'opération possède « plusieurs formes » et ad hoc par opposition au polymorphisme de sous-type utilisé lors d'une redéfinition ou au polymorphisme dit paramétrique utilisé dans les classes paramétrées (par un type).

système/system

A collection of connected units that are organized to accomplish a specific purpose. A system can be described by one or more models, possibly from different viewpoints.

T

transition

A relationship between two states indicating that an object in the first state will perform certain specified actions and enter the second state when a specified event occurs and specified conditions are satisfied. On such a change of state the transition is said to fire.

type (en JAVA)

In JAVA, a class is a type. An interface is a type. A primitive is a type. An array is a type. Therefore, every type is also either a class, an interface, a primitive, or an array. In addition, there are two distinct categories of types: primitive types and reference types. A variable of primitive type always holds a primitive value of that same type. Such a value can only be changed by assignment operations on that variable. A variable of reference type always holds the value of a reference to an object. All objects, including arrays, support the methods of class Object. The reference types are class types, interface types, and array types.

type abstrait/abstract type

Un type de données est un ensemble de valeurs muni d’opérations permettant de faire des calculs sur ces valeurs (en faisant parfois intervenir des valeurs d’autres types). Un type abstrait est indépendant de toute mise en œuvre; On se contente de donner le nom des ensembles de valeurs, sans dire comment celles-ci sont représentées, et l’en-tête des opérations, sans dire comment celles-ci seront réalisées. La notion de type abstrait est à l’origine des notions de classe et d’objet.
y a cependant des différences significatives: un type abstrait spécifie des ensembles de toutes les valeurs possibles alors qu'une classe correspond à un ensemble fini d'objets, qui peuvent être créées ou détruits, et auxquels on peut affecter des valeurs.

**Visibilité/Visibility**

An enumeration whose value (public, protected, private, or implementation) denotes how the model element to which it refers may be seen outside its enclosing name space.

**Vue/View**

A projection of a model, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective.