

Errata de la version 1.1 en date du 10/06/05 du document Règles de Cohérence UML 2.0

(date de l'errata : 17/08/05)

Note :

Quelques modifications ont été effectuées sur les règles de la section Classes de la version 1.1 du 10/06/05 du document Règles de Cohérence UML 2.0. Un tableau synthétique séparé (fichier ErrataV11_160805.pdf) reprend la liste des modifications. Cet errata en présente le détail. Lorsque la modification est minime (ne concerne qu'une remarque associée à la règle par exemple), le numéro de la règle est suivi d'une explication entre parenthèses indiquant la modification effectuée, qui est alors textuellement exprimée. Si la règle est entièrement reprise dans ce document d'errata, alors aucune explication n'est mentionnée entre parenthèses. Enfin, lorsque des règles sont supprimées ou rajoutées, cela est mentionné juste après le numéro de la règle dans la version 1.1 (pour les règles supprimées) ou après l'indication « REGLE - »(pour les nouvelles règles) avec la précision de l'endroit où elles ont été rajoutées pour la version suivante du document.

PARTIE ELEMENTS CLASSES

RÈGLE 13¹ :

L'expression d'une multiplicité doit suivre la syntaxe :

```
multiplicity ::= <multiplicity-range> [ '{ ' <order_designator>
                                     [ ' , ' <uniqueness-designator> ] }' ]
```

```
<multiplicity-range> ::= [ <lower> '.. ' ] <upper>
<lower> ::= <integer> | <value-specification>
<upper> ::= '*' | <value-specification>
<order-designator> ::= 'ordered' | 'unordered'
<uniqueness-designator> ::= 'unique' | 'nonunique'
```

RÈGLE 24 :

Les valeurs spécifiées par une contrainte doivent être de type booléen. [[7] p.54]

RÈGLE 31 : SUPPRIMÉE (devient un guide)

RÈGLE 58 : (Remarque rajoutée)

Remarque : Une opération sans effet de bord est représentée avec la propriété {query}.

RÈGLE 59 :

Une opération redéfinissant une autre opération doit respecter les propriétés suivantes : [[7] p.78]

- les deux opérations ont le même nombre de paramètres et le même nombre de résultats ;
- le type de chaque paramètre et résultat de l'opération qui redéfinit est

1 Nouvelle version d'Octobre 2004 de la superstructure d'UML 2.0 (dite « avec Issues »)

conforme au type du paramètre ou résultat de l'opération redéfinie, c'est-à-dire d'un même type ou d'un sous-type.

Remarque : La règle 59 est une redéfinition de la règle 49.

RÈGLE 60 : (Juste 2ième remarque modifiée)

Remarque : Deux opérations ont la même signature si et seulement si le nom des deux opérations est le même et les paramètres sont de mêmes types (en prenant en compte le nombre et l'ordre des types des paramètres).

RÈGLE 63 : (Remarque rajoutée)

Remarque : Une opération sans effet de bord est une opération à laquelle on a rajouté la propriété {query}.

RÈGLE 80 :

Les méthodes associées à des opérations ne doivent pas modifier d'attributs qui ont la propriété {readOnly}. [Nouvelle règle]

RÈGLE LIEN 4 : RAJOUTÉE (Après règle 81, dernière règle d' Attribute)

Tout attribut au sein d'une classe doit être unique. (cf règle 4).

RÈGLE - : RAJOUTÉE (Première règle de Signal)

Si un classificateur définit une réception, alors cette réception doit référencer un signal existant. [Nouvelle règle]

FIG 2-7 modifiée (Virgule rajoutée entre 2 paramètres génériques)

RÈGLE - : RAJOUTÉE (Dernière règle de Template Parameter > Classificateur)

Si le champ optionnel [`>` constraint] apparaît avec {contract}, alors le paramètre générique devra être substituable. [tirée de [7] p.557]

Remarque : Un classificateur est substituable à un autre s'il existe une dépendance de substitution ou si les classificateurs sont liés par une généralisation avec son méta-attribut {isSubstitutable} à vrai.

PARTIE RELATIONS CLASSES

RÈGLE 124 : (Remarque rajoutée)

Remarque : La visibilité de la relation d'importation est la visibilité obtenue par l'élément importé à l'intérieur de l'espace de nommage importateur.

RÈGLE 133 : SUPPRIMÉE (car redondante avec Règle 130)

RÈGLE 147 :

Le type de l'expression éventuellement utilisée pour décrire la dérivation doit être identique ou alors d'un sous-type du type de l'élément source.

[Nouvelle règle]

RÈGLE 148 :

Les types des éléments mis en jeu par une relation de dépendance de stéréotype « refine » doivent être conformes (c'est-à-dire que le type de la cible doit être du type ou d'un sous-type du type de la source). [Nouvelle règle]

RÈGLE - : RAJOUTÉE (Dernière règle de Substitution)

Dans le cas où la relation de substitution met en relation deux classes, il faut qu'aucune des deux classes ne soit abstraite. [Nouvelle règle]

RÈGLE 167 :

S'il existe une relation de composition entre les classes A et B (A étant la classe composée), alors l'objet créateur de l'instance d'une classe B doit être une instance de A ou d'une des classes filles de la classe A. [Nouvelle règle]

FIG 3-3 : modifiée (Trait pointillé Xor)

FIG 3-4 : modifiée (Multiplicités 0..1)

RÈGLE 195 : (Scindée en 2 règles)

- Une fin d'association navigable ne peut être redéfinie que par une fin d'association navigable. [[7] p.91]
- Une fin d'association navigable ne peut être le sur-ensemble que d'une fin d'association navigable. [[7] p.91]

RÈGLE 197 : (Remarque enlevée)

Lorsqu'une fin d'association possède un qualificateur, l'autre fin doit être navigable. [Nouvelle Règle]

RÈGLE 198 : (Origine de la règle changée²)

RÈGLE 199 : SUPPRIMÉE ³

RÈGLE 203 : SUPPRIMÉE ⁴

2 Etait une nouvelle règle de la V 1.1, mais provient maintenant de manière sûre de la nouvelle version d'Octobre 2004 de la superstructure d'UML 2.0 (dite « avec Issues »)

3 N'est plus valide dans la nouvelle version de la superstructure d'UML 2.0 avec Issues

4 N'est plus valide dans la nouvelle version de la superstructure d'UML 2.0 avec Issues