



**ENTERPRISE ARCHITECT**

Série de Guides d'Utilisateur

# Validation Modèle

Author: Sparx Systems

Date: 7/11/2024

Version: 17.0

CRÉÉ AVEC  **ENTERPRISE  
ARCHITECT**

## Table des Matières

Validation Modèle	3
Configurer la validation Modèle	5
Validation Exécuter	6
Référence aux règles	7
Bien-formé	9
Composition des éléments	11
Validité de la propriété	12
Conformité OCL	13

# Validation Modèle



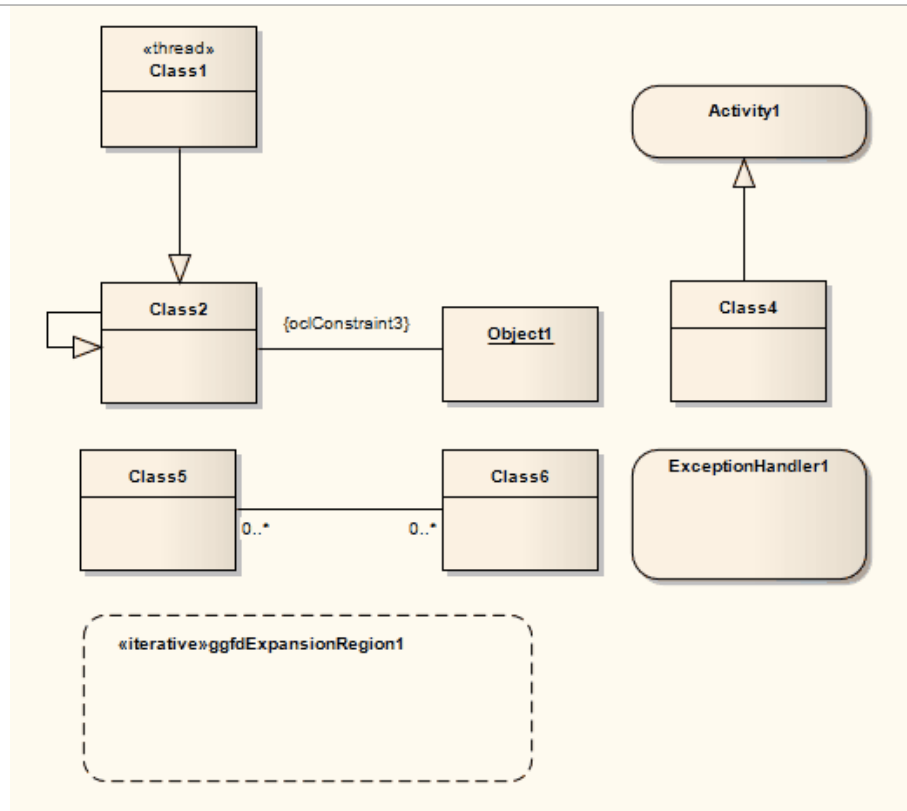
Vous utilisez Modèle Validation pour vérifier les modèles UML par rapport aux règles UML connues. L'ensemble de règles à valider peut être personnalisé en activant et en désactivant certaines règles et catégories de règles. Vous pouvez exécuter Modèle Validation sur un seul élément UML, un diagramme ou un Paquetage entier. Vous pouvez compléter les règles de validation fournies par le système en utilisant un Add-In ou une MDG Technologie (comme BPMN) qui contient ses propres règles. add-ins basés sur Modèle sont un outil efficace pour créer des ensembles de règles nouveaux et spécifiques qui implémentent des règles personnalisées spécifiquement liées à un projet, un client ou un domaine unique.

## Accéder

Ruban	Conception > Paquetage > Gérer > Valider > Valider Paquetage actuel
-------	---------------------------------------------------------------------

## Concepts de validation

Concept	Détail
Validation des objets	Valider un UML : <ul style="list-style-type: none"> <li>• Element valide l'élément et ses enfants, ses fonctionnalités (attributs et opérations) et ses relations (connecteurs)</li> <li>• Diagramme valide le diagramme lui-même (pour son exactitude) ainsi que tous les éléments et connecteurs au sein du diagramme</li> <li>• Paquetage valide le Paquetage et tous les sous-Packages, éléments, connecteurs et diagrammes qu'il contient</li> </ul>
Exemple - Violation Modèle	Ce diagramme UML contient plusieurs violations fondamentales des règles UML :



Si vous exécutez la validation Modèle sur ce diagramme, ces violations sont identifiées dans la fenêtre de sortie du système :

- Une région d'expansion UML (ExpansionRegion1) n'a pas son entrée enfant ExpansionNode
- Une auto-généralisation invalide existe sur Class2 (les éléments UML ne peuvent pas être auto-généralisés)
- Une violation OCL existe pour l'association anonyme (entre Class2 et Object1)
- Un gestionnaire d'exceptions UML (ExceptionHandler1) n'a pas son ObjectNode d'entrée enfant

## Configurer la validation Modèle

Avant d'effectuer une validation de modèle, vous devez activer ou désactiver les règles appliquées par le validateur de modèle. Vous pouvez définir des règles supplémentaires à partir de tout Add-Ins supplémentaire pouvant être installé en plus d' Enterprise Architect .

### Accéder

Ruban	Conception > Paquetage > Gérer > Outils > Valider > Configurer les règles de validation
-------	-----------------------------------------------------------------------------------------

### Configurer la validation

Dans la dialogue « Configuration de la validation Modèle », cliquez sur la case à cocher en regard de chaque règle de validation à appliquer pour effectuer une validation de modèle. Cliquez sur le bouton OK pour définir les règles de validation sélectionnées.

# Validation Exécuter

Après avoir configuré la validation que vous souhaitez effectuer et défini les règles de validation à appliquer, vous pouvez exécuter la validation sur les objets que vous avez sélectionnés soit dans la fenêtre Navigateur , soit dans un diagramme . Si vous ne souhaitez pas poursuivre une validation que vous avez commencée, vous pouvez l'annuler.

## Accéder

Ruban	Conception > Paquetage > Gérer > Valider > Valider Paquetage actuel
Raccourcis Clavier	Ctrl+Alt+V

## Validation

Pendant l'exécution de la validation, une fenêtre de progression s'affiche. Une fois la validation terminée, les résultats s'affichent dans la fenêtre Sortie système. Vous pouvez mettre en surbrillance et sélectionner sur un diagramme l'élément auquel un message d'erreur fait référence, en double-cliquant sur la ligne d'erreur dans la fenêtre Sortie système.

## Annuler la validation

À tout moment lors d'une validation, soit :

- Cliquez sur le bouton Annuler la validation dans la fenêtre de progression de la validation, ou
- Sélectionnez « Conception > Paquetage > Gérer > Valider > Annuler la validation en cours »

## Notes

- Si la fenêtre Sortie système ne s'affiche pas automatiquement, sélectionnez « Démarrer > Toutes Windows > Conception > Explorer > Système ».

## Référence aux règles

La validation Modèle fonctionne en appliquant un ensemble de règles de validation à l'objet sélectionné. Ces règles sont organisées en groupes.

### Groupes de règles

Groupe	Description
Bien-formé	<p>Le validateur applique ces règles pour vérifier si un élément, une relation, fonctionnalité ou diagramme est bien formé ; par exemple, si l'objet est un élément UML valide ou si un diagramme contient des éléments valides.</p> <p>Comme expliqué dans <i>Format des messages d'erreur</i>, ces règles génèrent des messages avec les codes :</p> <ul style="list-style-type: none"><li>• MVR01... (Élément : case à cocher Bien formé sélectionnée)</li><li>• MVR05... (Relation : case à cocher « Bien formé » sélectionnée)</li><li>• MVR08... (Fonctionnalité : case à cocher de bonne formation sélectionnée)</li><li>• MVR0B... (Diagramme : case à cocher de bonne formation sélectionnée)</li></ul>
Composition des éléments	<p>Le validateur applique ces règles pour vérifier si un élément UML contient ou non des enfants valides, s'il contient le bon nombre d'enfants valides et si l'élément manque ou non d'enfants requis.</p> <p>Ces règles génèrent des messages avec le code :</p> <ul style="list-style-type: none"><li>• MVR02... (Élément : case à cocher Composition sélectionnée)</li></ul>
Validité de la propriété	<p>Le validateur applique ces règles pour vérifier si l'élément, la relation ou fonctionnalité possède ou non les propriétés UML correctes définies et si les propriétés contiennent des valeurs incorrectes ou conflictuelles.</p> <p>Ces règles génèrent des messages avec les codes :</p> <ul style="list-style-type: none"><li>• MVR03... (Élément : case à cocher Validité de la propriété sélectionnée)</li><li>• MVR06... (Relation : case à cocher Validité de la propriété sélectionnée)</li><li>• MVR09... (Fonctionnalité : case à cocher Validité de la propriété sélectionnée)</li></ul>
Propriétés personnalisées	<p>Le validateur applique ces règles pour vérifier un élément, une relation ou fonctionnalité par rapport à toutes les contraintes définies dans OCL.</p> <p>Ces règles génèrent des messages avec les codes :</p> <ul style="list-style-type: none"><li>• MVR04... (Élément : case à cocher Conformité OCL sélectionnée)</li><li>• MVR07... (Relation : case à cocher Conformité OCL sélectionnée)</li><li>• MVR0A... (Fonctionnalité : Case Conformité OCL cochée)</li></ul>

### Format du message d'erreur

Lorsque vous effectuez une validation, toutes les violations des règles sont répertoriées dans la fenêtre Sortie système. Chaque violation est associée à un ID de violation au format :

MVRxxxxnn

- MVR signifie Modèle Validation Rule
- xx est un nombre hexadécimal correspondant à la position de la règle de validation dans la dialogue « Configuration de validation Modèle », indiquant ainsi quelle règle est appliquée et violée
- nnnn est le numéro du message de violation

Par exemple:

- Les messages avec l' ID MVR01nnnn indiquent que la case à cocher « Élément : Bien formé » est sélectionnée et qu'une violation de cette règle a été détectée
- Les messages avec l' ID MVR0Annnn indiquent que la case à cocher « Fonctionnalité : Conformité OCL » (10e dans l'ordre dans le dialogue , ou Ath en hexadécimal) est sélectionnée et qu'une violation de cette règle a été détectée



## Bien-formé

Les règles de bonne formation permettent de vérifier si un élément, une relation, fonctionnalité ou diagramme est bien formé. Par exemple, elles permettent de vérifier si l'élément cible est un élément UML valide ou si un diagramme contient des éléments valides.

### Violations signalées

ID de violation	Description
MVR010001	« Element » n'est pas un élément UML valide Erreur : l'élément n'est pas un élément UML 2.5 reconnu.
MVR050001	«Relation» n'est pas une relation UML valide Erreur : la relation n'est pas une relation UML 2.5 reconnue.
MVR050002	« Relation » n'est pas légal pour « Élément de départ » --> « Élément de fin » Erreur : la relation entre les éléments de début et de fin donnés n'est pas valide pour ces éléments.
MVR050003	« Élément parent » : isLeaf=true et ne peut pas être généralisé par « Élément enfant » Erreur : la relation de généralisation ne peut pas exister entre les éléments parent et enfant car l'élément parent est défini comme un élément feuille.
MVR050004	« Élément enfant » : isRoot=true et ne peut pas généraliser « Élément parent » Erreur : la relation de généralisation ne peut pas exister entre les éléments parent et enfant car l'élément enfant est défini comme un élément racine.
MVR050005	« L'élément » ne peut pas se généraliser lui-même Erreur : l'élément ne peut pas être auto-généralisé.
MVR050006	Le connecteur d'imbrication ne correspond pas à l'imbrication dans Navigateur Avertissement : l'élément source du connecteur imbriqué n'est pas imbriqué sous l'élément cible dans l'arborescence du projet.
MVR0B0001	Violation du diagramme d'états : « informations étendues » Erreur : le diagramme State contient une violation UML ; les « informations étendues » fournissent plus de détails sur la violation spécifique.
MVR0C0001	Violation Séquence : « informations étendues » Erreur : Le diagramme Séquence contient une violation, les « informations étendues » fournissent plus de détails sur la violation spécifique. (Cette règle valide les opérations appelées par Messages - telles que sélectionnées dans la liste déroulante de la boîte dialogue « Propriétés des messages » - et signalera une violation si l'opération n'appartient pas à Lifeline.)
MVR7F0001	«Exigence» non réalisée Erreur : un élément d'exigence doit se trouver à l'extrémité cible d'un connecteur

	de réalisation, d'association (y compris l'agrégation et la composition) ou de généralisation.
--	------------------------------------------------------------------------------------------------

## Composition des éléments

Les règles de composition d'éléments sont appliquées pour vérifier si un élément UML contient ou non des enfants valides, s'il contient le bon nombre d'enfants valides et si l'élément manque ou non d'enfants requis.

### Violations signalées

ID d'erreur	Description
MVR020001	« Élément » ne dispose pas de l'élément enfant requis « Élément enfant » Information : l'élément manque d'un élément enfant de type Élément enfant.
MVR020002	Paquetage UML enfant non valide Information : L'élément ne peut pas être un enfant direct du Paquetage et doit être un enfant d'un autre élément (par exemple, les ports doivent être des enfants d'autres éléments et non des membres directs Paquetage UML ).
MVR020003	Enfant non valide « Nom de l'élément enfant » (« Type d'élément enfant ») Information : l'élément enfant n'est pas valide sur l'élément parent testé.

## Validité de la propriété

Les règles de validité des propriétés sont appliquées pour vérifier si un élément, une relation ou fonctionnalité possède ou non les propriétés UML correctes définies pour lui et si elles contiennent des valeurs incorrectes ou conflictuelles.

### Violations signalées

ID d'erreur	Description
MVR030001	La propriété « Élément » : « Propriété » n'est pas définie Information : La propriété element ne contient aucune valeur .
MVR030002	« Élément » : la propriété « Propriété » a valeur non valide : « Valeur » Information : La propriété de l'élément contient une valeur non valide.
MVR030003	« Élément » : isLeaf=true et ne peut pas être abstrait Information : les propriétés isLeaf et isAbstract de l'élément sont toutes deux définies sur True, ce qui n'est pas valide.
MVR060001	La propriété « Relation » : « Propriété » n'est pas définie Information : La propriété de relation ne contient aucune valeur .
MVR060002	« Relation » : la propriété « Propriété » a valeur non valide : « Valeur » Information : la propriété de relation contient une valeur non valide.
MVR090001	Non-concordance Attribut/AssociationEnd, « Attribut » : « Description de la non-concordance », ... Information : l'attribut donné a une associationEnd du même nom mais ils diffèrent dans les détails répertoriés.

## Conformité OCL

Les règles de conformité OCL sont appliquées pour valider un élément, une relation ou un attribut par rapport à des contraintes définies dans le langage OCL ( Object Constraint Language). OCL est utilisé pour décrire des expressions sur des modèles UML et pour exprimer des contraintes sans effets secondaires. Vous pouvez ajouter des contraintes OCL à n'importe quel élément, relation ou attribut dans Enterprise Architect .

### Définir les contraintes OCL

Vous pouvez ajouter une contrainte OCL à un objet modèle en définissant OCL comme type de contrainte dans la dialogue « Propriétés » objet .

Pour	Détail
Un élément	<p>Cliquez-droit sur l'élément et sélectionnez ' Propriétés   Propriétés Dialogue '. Cliquez sur la page 'Contraintes'; dans le champ ' Type ' cliquez sur la flèche déroulante et sélectionnez OCL.</p> <p>Pour effectuer une validation OCL :</p> <ol style="list-style-type: none"> <li>Sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Outils &gt; Valider &gt; Configurer les règles de validation » et cochez la case « Élément : Conformité (OCL) ».</li> <li>Cliquez sur l'élément et sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Outils &gt; Valider &gt; Valider Paquetage actuel ».</li> </ol> <p>Toutes les violations de syntaxe OCL sont enregistrées dans l'onglet « Validation Modèle » de la fenêtre Sortie système.</p>
Une relation	<p>Sur un diagramme , cliquez-droit sur le connecteur et sélectionnez l'option ' Propriétés '; sélectionnez la page 'Contraintes', et dans le champ ' Type ' cliquez sur la flèche déroulante et sélectionnez 'OCL'.</p> <p>Pour effectuer une validation OCL :</p> <ol style="list-style-type: none"> <li>Sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Outils &gt; Valider &gt; Configurer les règles de validation » et cochez la case « Relation : Conformité (OCL) »</li> <li>Cliquez sur le connecteur et sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Outils &gt; Valider &gt; Valider Paquetage actuel »</li> </ol> <p>Toutes les violations de syntaxe OCL sont enregistrées dans l'onglet « Validation Modèle » de la fenêtre Sortie système.</p>
Un attribut	<p>Cliquez-droit sur l'élément et sélectionnez ' Attributes   Contraintes'; dans le champ ' Type ', cliquez sur la flèche déroulante et sélectionnez 'OCL'.</p> <p>Pour effectuer une validation OCL :</p> <ol style="list-style-type: none"> <li>Sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Valider &gt; Configurer les règles de validation » et cochez la case « Fonctionnalité : Conformité (OCL) ».</li> <li>Cliquez sur l'élément et sélectionnez « Conception &gt; Paquetage &gt; Gérer &gt; Valider &gt; Valider Paquetage actuel ».</li> </ol> <p>Toutes les violations de syntaxe OCL sont enregistrées dans l'onglet « Validation Modèle » de la fenêtre Sortie système.</p>

## Violations signalées

ID d'erreur	Description
MVR040001	Violation de l'OCL : « OCL violé » Information : la contrainte OCL de l'élément utilise une syntaxe incorrecte.
MVR070001	Violation de l'OCL : « OCL violé » Information : la contrainte OCL de la relation utilise une syntaxe incorrecte.
MVR0A0001	Violation de l'OCL : « OCL violé » Information : la contrainte OCL de l'attribut utilise une syntaxe incorrecte.

## Notes

- Pour avoir une contrainte OCL valide, la syntaxe doit être correctement formée ; si l'expression n'est pas correcte, un message s'affiche indiquant que la contrainte OCL n'est pas valide

