Enterprise Architect

Model Validation

Perform Model Validation in Sparx Systems Enterprise Architect, to check models against UML rules or constraints defined using Object Constraint Language (OCL), against an element, diagram or Package. Add other rules from an Add-In or MDG Technology.

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Model Validation

You use Model Validation to check UML models against known UML rules (which you identify when configuring validation) as well as any constraints defined within the model using the Object Constraint Language (OCL). You can run Model Validation against a single UML element, a diagram or an entire Package.

You can add to the system-provided validation rules by using an Add-In or an MDG Technology (such as BPMN) that contains its own rules.
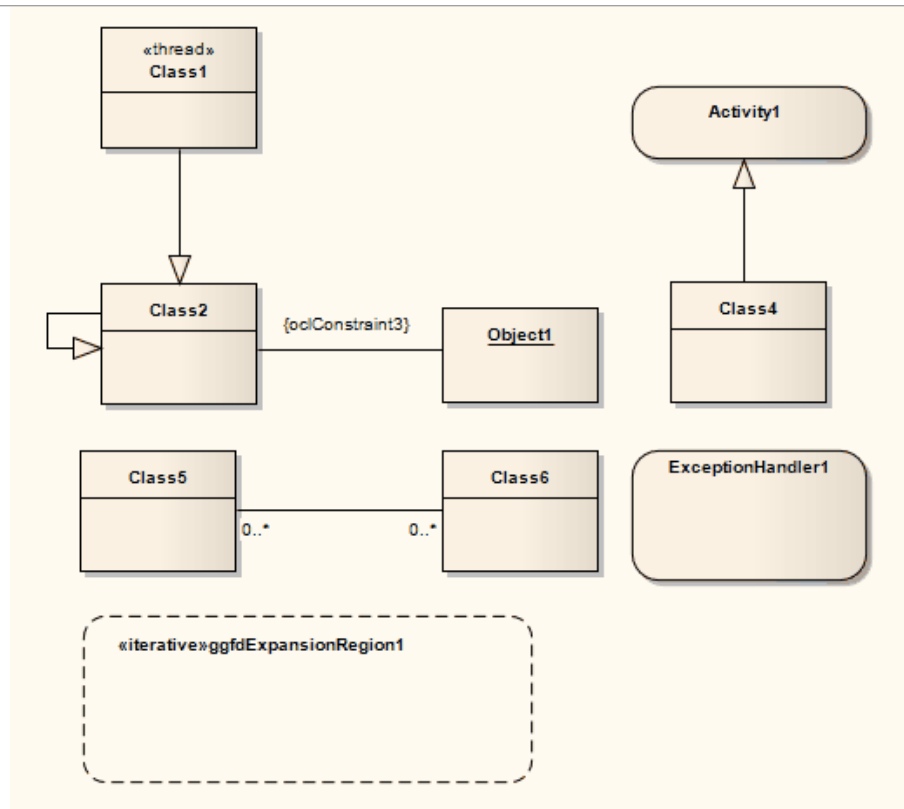
Access

Ribbon	Design > Model > Manage > Validate > Validate Current Package

Validation Concepts

Concept	Detail

Validating Objects	Validating a UML: • Element validates the element and its children, its features (attributes and operations) and its relationships (connectors) • Diagram validates the diagram itself (for correctness) as well as any elements and connectors within the diagram • Package validates the Package and all sub-Packages, elements, connectors and diagrams within it
Example - Model Violation	This UML diagram contains several basic violations of the UML rules:

If you run Model Validation on this diagram, these violations are identified in the System Output window:

- A UML ExpansionRegion (ExpansionRegion1) is missing its child input ExpansionNode

- An invalid self-generalization exists on Class2 (UML elements cannot be self-generalized)

- An OCL violation exists for the anonymous Association (between Class2 and Object1)

- A UML ExceptionHandler (ExceptionHandler1) is missing its

	child input ObjectNode

Configure Model Validation

Before you perform a model validation, you enable or disable the rules that are applied by the model validator. You can define additional rules from any additional Add-Ins that might be installed besides Enterprise Architect.

Access

Ribbon	Design > Model > Manage > Validate > Configure Validation Rules

Configure Validation

On the 'Model Validation Configuration' dialog, click on the checkbox against each Validation Rule to apply in performing a model validation. Click on the OK button to set the validation rules selected.

Run Validation

After you have configured the validation you want to perform and set the validation rules to be applied, you can execute the validation on objects you have selected from either the Browser window or a diagram. If you do not want to continue a validation you have started, you can cancel it.

Access

Ribbon	Design > Model > Manage > Validate > Validate Current Package
Keyboard Shortcuts	Ctrl+Alt+V

Validation

While the validation executes it displays a progress window. When validation is complete, the results are displayed in the System Output window. You can highlight and select on a diagram the element that an error message refers to, by double-clicking on the error line in the System Output window.

Cancel Validation

At any time during a validation, either:

- Click on the Cancel Validation button on the validation progress window, or
- Select 'Design > Model > Manage > Validate > Cancel Current Validation'

Notes

- If the System Output window does not automatically display, select 'Start > Desktop > Design > System Output'.

Rules Reference

Model Validation works by applying a set of validation rules to the selected object. These rules are arranged in groups.

Rule Groups

Group	Description
Well-Formed ness	The validator applies these rules to check whether or not an element, relationship, feature or diagram is well-formed; for example, whether the object is a valid UML item or whether a diagram contains valid elements within it. As explained in *Error Message Format*, these rules generate messages with the codes: • MVR01... (Element: Well Formedness checkbox selected) • MVR05... (Relationship: Well Formedness checkbox selected) • MVR08... (Feature: Well Formedness checkbox selected) • MVR0B... (Diagram: Well Formedness checkbox selected)

Element Composition	The validator applies these rules to check whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children. These rules generate messages with the code: • MVR02... (Element: Composition checkbox selected)
Property Validity	The validator applies these rules to check whether or not the element, relationship or feature has the correct UML properties defined, and whether the properties contain incorrect or conflicting values. These rules generate messages with the codes: • MVR03... (Element: Property Validity checkbox selected) • MVR06... (Relationship: Property Validity checkbox selected) • MVR09... (Feature: Property Validity checkbox selected)
Custom Properties	The validator applies these rules to check an element, relationship or feature against

	any defined constraints in OCL.
	These rules generate messages with the codes:
	• MVR04... (Element: OCL Conformance checkbox selected)
	• MVR07... (Relationship: OCL Conformance checkbox selected)
	• MVR0A... (Feature: OCL Conformance checkbox selected)

Error Message Format

When you perform a validation, any violations of the rules are listed on the System Output window. Each violation has a violation ID of the format:

MVRxxnnnn

• MVR stands for Model Validation Rule

• xx is a hexadecimal number corresponding to the position of the validation rule in the 'Model Validation Configuration' dialog, thus indicating which rule is applied and violated

• nnnn is the number of the violation message

For example:

• Messages with the ID MVR01nnnn indicate that the 'Element: Well-Formedness' checkbox is selected and a

violation of that rule has been detected

- Messages with the ID MVR0Annnn indicate that the 'Feature: OCL Conformance' checkbox (10th in order on the dialog, or Ath in hexadecimal) is selected and a violation of that rule has been detected

Well-Formedness

Well-formedness rules are applied to check whether or not an element, relationship, feature or diagram is well-formed. For example, whether the target item is a valid UML item, or whether a diagram contains valid elements within it.

Reported violations

Violation ID	Description
MVR010001	«Element» is not a valid UML Element Error: The element is not a recognized UML 2.5 element.
MVR050001	«Relationship» is not a valid UML Relationship Error: The relationship is not a recognized UML 2.5 relationship.
MVR050002	«Relationship» is not legal for «Start Element» --> «End Element» Error: The relationship between the given start and end elements is not valid for those elements.
MVR050003	«Parent Element»:isLeaf=true and cannot

	be generalized by «Child Element» Error: The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element.
MVR050004	«Child Element»:isRoot=true and cannot generalize «Parent Element» Error: The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element.
MVR050005	«Element» cannot generalize self Error: The element cannot be self-generalized.
MVR0B0001	Statechart violation: «extended information» Error: The State diagram contains a UML violation; «extended information» provides more details on the specific violation.
MVR0C0001	Sequence Violation: «extended information» Error: The Sequence diagram contains a violation, «extended information»

	provides more details about the specific violation.
	(This rule validates operations being called by Messages - as selected from the drop down combo box of the 'Messages Properties' dialog - and will report a violation if the operation does not belong to the Lifeline.)
MVR7F0001	«Requirement» Not realized
	Error: A Requirement element must be at the target end of a Realization, Association (including Aggregation and Composition) or Generalization connector.

Element Composition

Element Composition rules are applied to check whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.

Reported violations

Error ID	Description
MVR020001	«Element» is missing required child element «Child Element» Information: The element is missing a child element of type Child Element.
MVR020002	Invalid UML Package child Information: The element cannot be a direct child of the Package and must be a child of another element (for example, Ports must be children of other elements, and not direct UML Package members).
MVR020003	Invalid child «Child Element name» («Child Element Type») Information: The child element is invalid on the tested parent element.

Property Validity

Property Validity rules are applied to check whether or not an element, relationship or feature has the correct UML properties defined for it and whether they contain incorrect or conflicting values.

Reported violations

Error ID	Description
MVR030001	«Element»:«Property» property is undefined Information: The element property contains no value.
MVR030002	«Element»:«Property» property has invalid value: "«Value»" Information: The element property contains an invalid value.
MVR030003	«Element»:isLeaf=true and cannot be abstract Information: The element's isLeaf and isAbstract properties are both set to True, which is invalid.

MVR060001	«Relationship»:«Property» property is undefined
	Information: The relationship property contains no value.
MVR060002	«Relationship»:«Property» property has invalid value: "«Value»"
	Information: The relationship property contains an invalid value.
MVR090001	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description», ...
	Information: The given attribute has an associationEnd of the same name but they differ in the listed details.

OCL Conformance

OCL Conformance rules are applied to validate an element, relationship or attribute against any defined constraints in the Object Constraint Language (OCL). OCL is used to describe expressions on UML models, and to express constraints free of side-effects. You can add OCL constraints to any element, relationship or attribute in Enterprise Architect.

Define OCL Constraints

You can add an OCL constraint to a model object by setting OCL as the Constraint type in the object 'Properties' dialog.

For	Detail
An Element	Right-click on the element and select 'Properties | Properties Dialog'. Click on the 'Constraints' page; in the 'Type' field click on the drop-down arrow and select OCL.
	To perform an OCL Validation:
	1. Select 'Design > Model > Manage > Validate > Configure Validation Rules' and select the 'Element: (OCL) Conformance' checkbox.
	2. Click on the element and select 'Design

	> Model > Manage > Validate > Validate Current Package'. Any OCL syntax violations are recorded in the 'Model Validation' tab of the System Output window.
A Relationship	On a diagram, right-click on the connector and select the 'Properties' option; select the 'Constraints' page, and in the 'Type' field click on the drop-down arrow and select 'OCL'. To perform an OCL Validation: 1. Select 'Design > Model > Manage > Validate > Configure Validation Rules' and select the 'Relationship: (OCL) Conformance' checkbox 2. Click on the connector and select 'Design > Model > Manage > Validate > Validate Current Package' Any OCL syntax violations are recorded in the 'Model Validation' tab of the System Output window.
An Attribute	Right-click on the element and select 'Attributes | Constraints'; in the 'Type' field click on the drop-down arrow and select 'OCL'. To perform an OCL Validation:

| | 1. Select 'Design > Model > Manage > Validate > Configure Validation Rules' and select the 'Feature: (OCL) Conformance' checkbox.

2. Click on the element and select 'Design > Model > Manage > Validate > Validate Current Package'.

Any OCL syntax violations are recorded in the 'Model Validation' tab of the System Output window. |

Reported violations

Error ID	Description
MVR040001	OCL violation: «violated OCL» Information: The element's OCL constraint uses incorrect syntax.
MVR070001	OCL violation: «violated OCL» Information: The relationship's OCL constraint uses incorrect syntax.
MVR0A0001	OCL violation: «violated OCL» Information: The attribute's OCL

	constraint uses incorrect syntax.

Notes

- To have a valid OCL constraint, the syntax must be correctly formed; if the expression is not correct, a message displays stating that the OCL constraint is not valid