JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Object Recognition for Compliance, Usability, and Sustainment

Patrick Morrison
Deandre Dixon
Justin Coe

orcus@jhuapl.edu

Outline

- Motivation and Benefits of ORCUS
- Defining ORCUS Meta-Models
- Using ORCUS in Cameo 2021x for Active Validation
- Using ORCUS in Cameo 2021x for Discrete Validation
- Summary

Motivation to Create ORCUS

- Lack of effective methods to verify compliance against program-specific meta-models
- Review is mostly a manual process of SME's and modelers sifting through meta-models and comparing information
  - ORCUS provides a more reactive and immediate framework for review
- Current need to establish similarities between models to improve cross-organization model readability and interoperability
- Facilitate integration of independent models from multiple organizations
  - Saves time, cost, and rework
- Need to facilitate faster modeling and decrease time to adoption for new Cameo users

What is ORCUS?

- ORCUS is a JHU APL-developed Cameo plug-in that is integrated with OpenAPI
- ORCUS utilizes a meta-model to establish validation patterns for evaluating models
  - **Meta-model agnostic** – no rework of ORCUS is needed to use with new models
  - Minimal rework to make existing meta-models readable by ORCUS using pre-defined stereotypes
  - Capable of comparing against multiple meta-models simultaneously
- ORCUS Provides several user interactive methods to identify and resolve violations in compliance with selected meta-models

Benefits of ORCUS

- **Ensures model compliance** and enables **faster development** for both new and experienced Cameo users
  - Verifies model compliance while building the model
- Enforces good modeling practices and patterns across an organization's modeling ecosystem
- **Provides metrics** to measure model compliance to program meta-models
  - Built-in validation has some capability, but does not compare against program-specific standards
  - Allows review of models to progress much faster and with more readily-available metrics

ORCUS

Benefits of ORCUS

- Continuous integration to **manage model federation**
- Able to **easily distribute** ORCUS-ready meta-models across programs
  - Meta-model only needs to be edited once to conform to ORCUS rules and can be shared among users
  - ORCUS uses a flexible platform-independent caching mechanism to promote reusability
  - Allows users to share meta-models quickly and ensures consistency across your program

Active vs Discrete Validation

Active Validation	Discrete Validation
• Intended for **new models** to ensure compliance from the start or when working with **existing models** to make them compliant • Displays violations as the user models • Provides additional method to remedy violations via ORCUS suggested compliance pop-up • Allows for drawing new relationships on diagram • Multiple display formats available to user • Full-diagram validation	• Intended for **snapshots of model** to measure progress towards improving model compliance. • Exports list of all violations to CSV or JSON • Provides all suggested actions to remedy violations with export • Allows for flagging violations in the model • Provides full-model validation and basic metrics to measure model compliance

What is a meta-model?

- “A model of/about a model.”

- Meta-models are used for establishing and conveying standard modeling patterns to be followed
  - Explain the model structure in terms of element types and relationships
  - Manually learned and adhered to by modelers
- Important to adhere to standard model patterns
  - Enable consistent querying and analysis throughout the model
  - Facilitate collaboration, especially when integrating disparate models

Simple Example Meta-Model

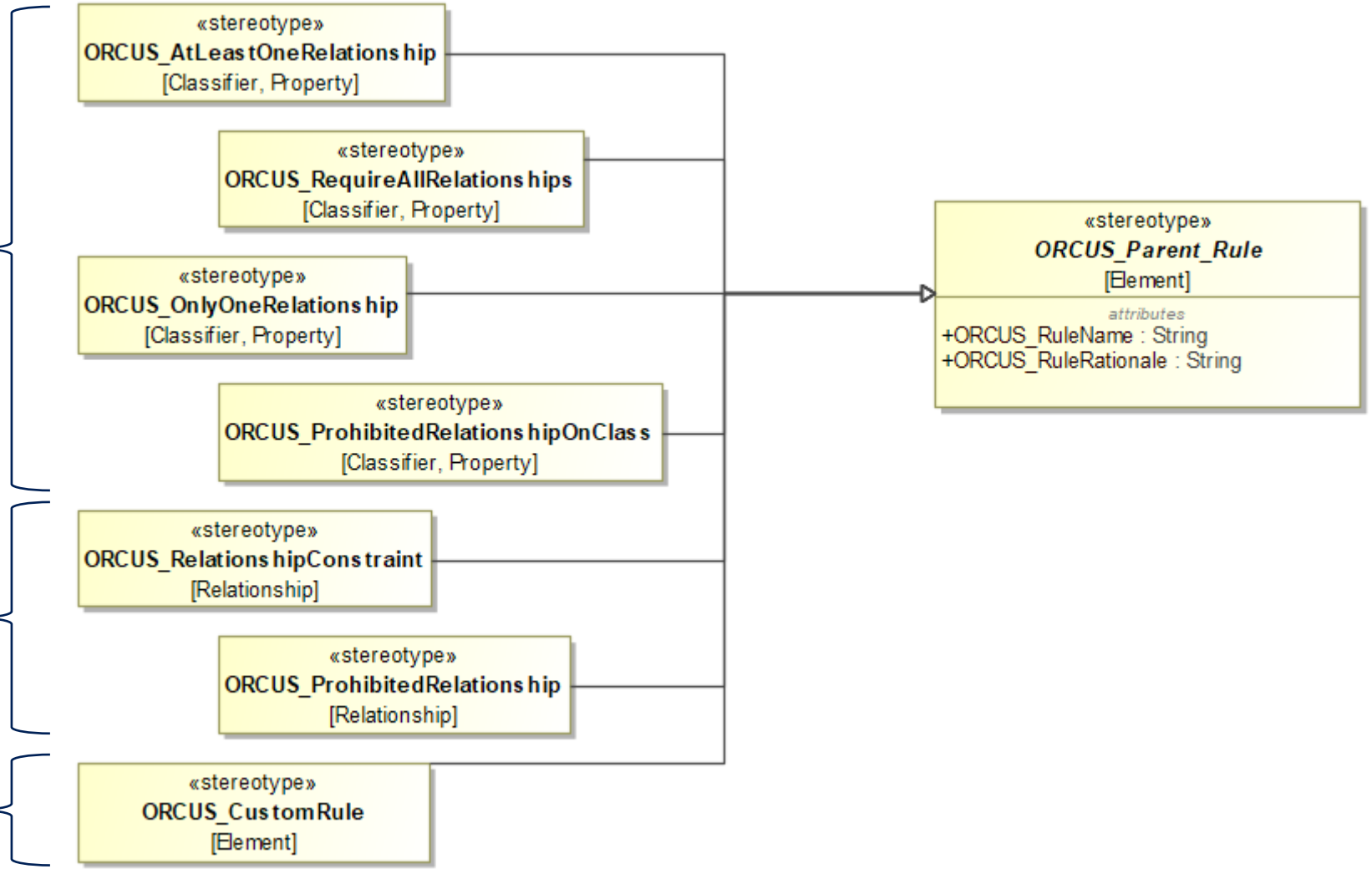
- Activities **Satisfy** Requirements
- Activities are **Allocated to** Actors

Defining ORCUS Meta-Model with Stereotypes

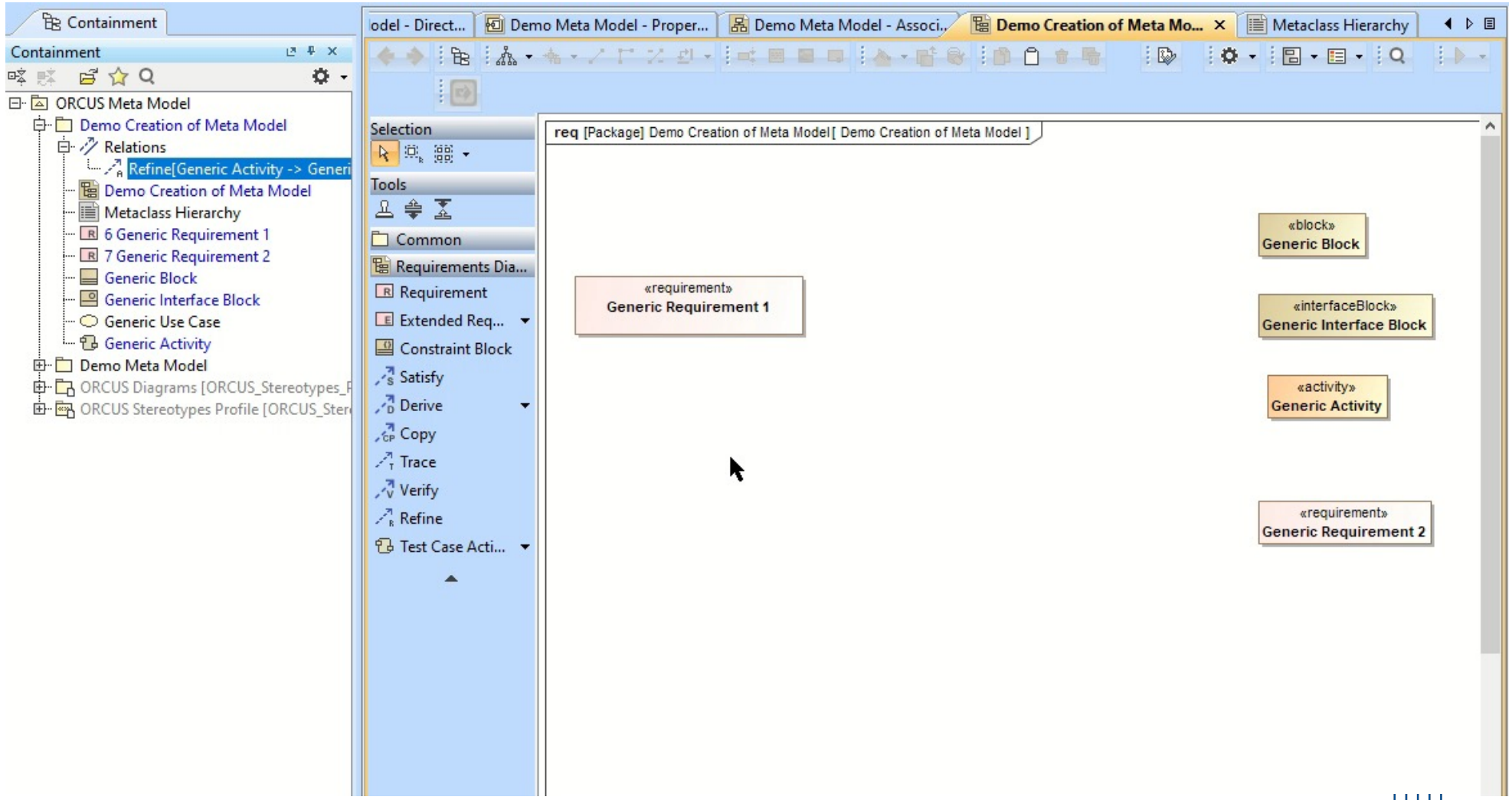
Base Rules

Relationship Rules

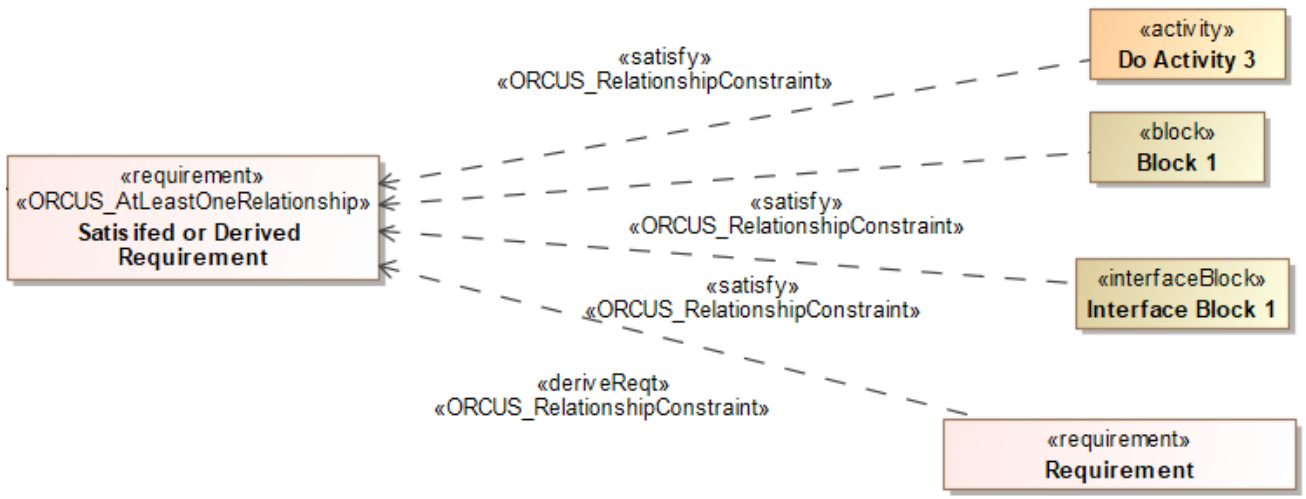
Customized Rules



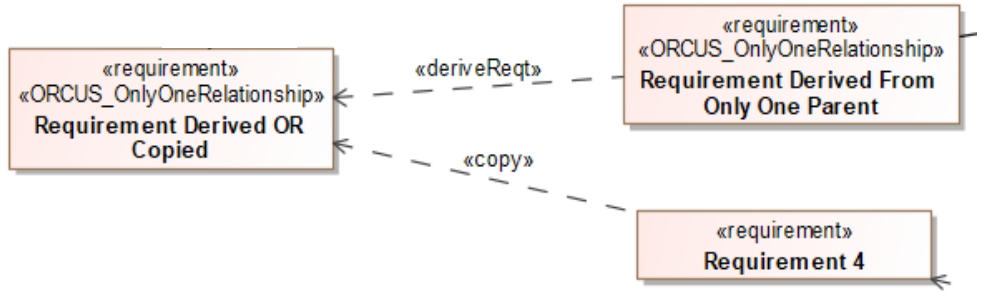
How to Define an ORCUS Meta-Model



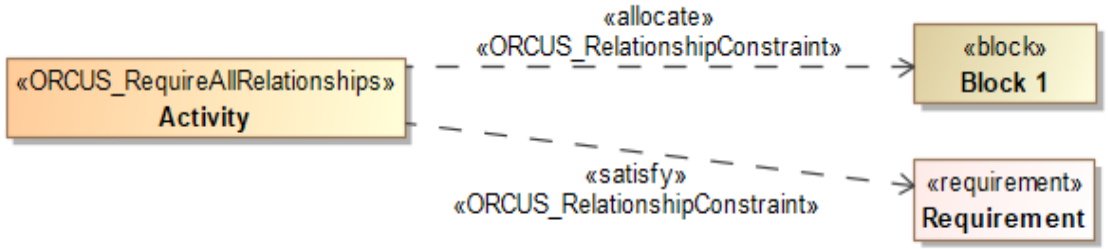
Meta-Model Rules Defined for this Demo



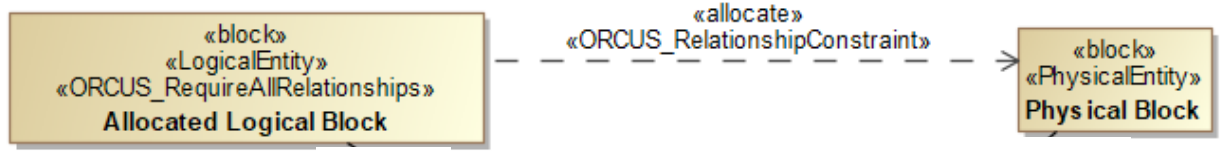
"A requirement must be satisfied or derived."



"A requirement can be copied or derived, but not both."

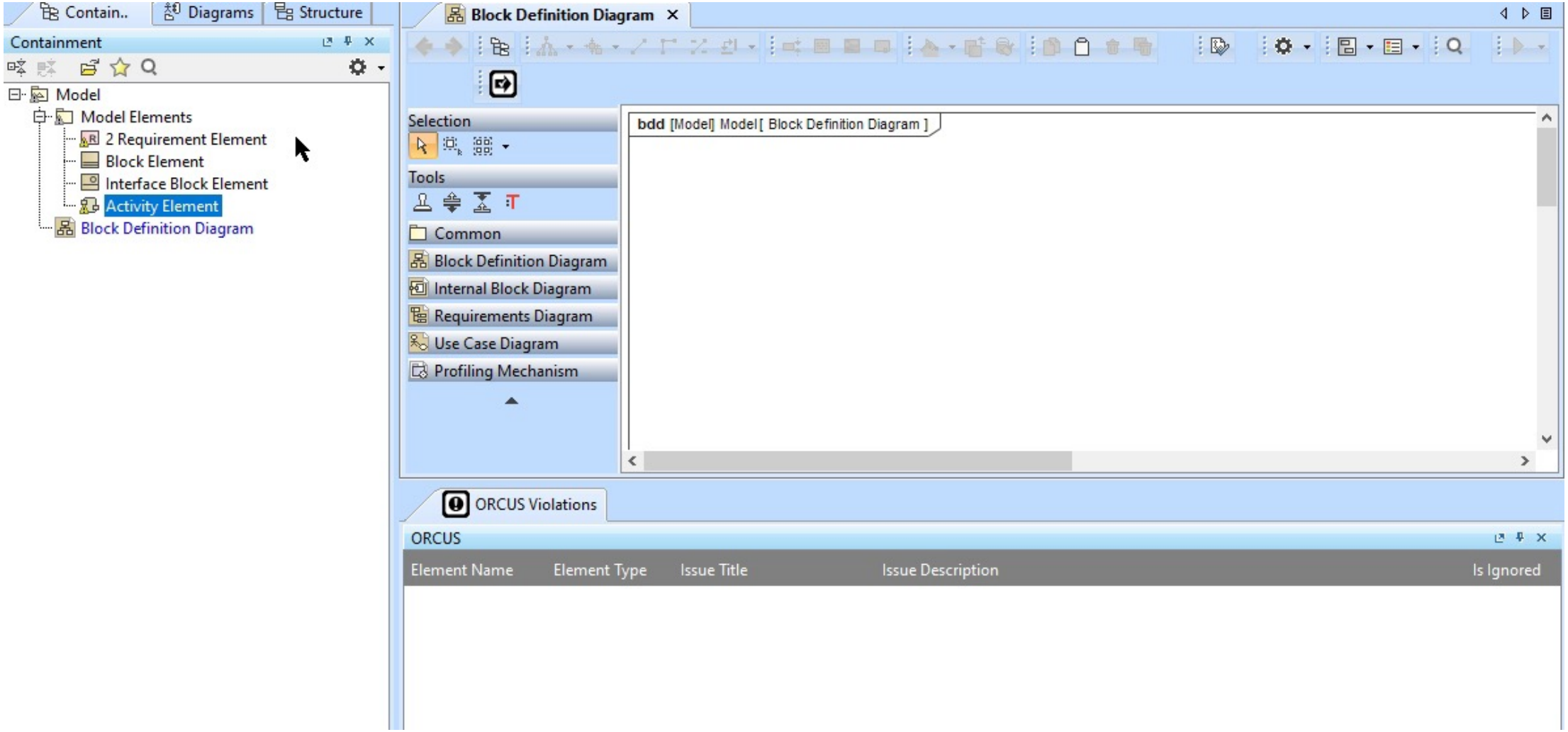


"An activity must be allocated to a block.
An activity must satisfy a requirement."

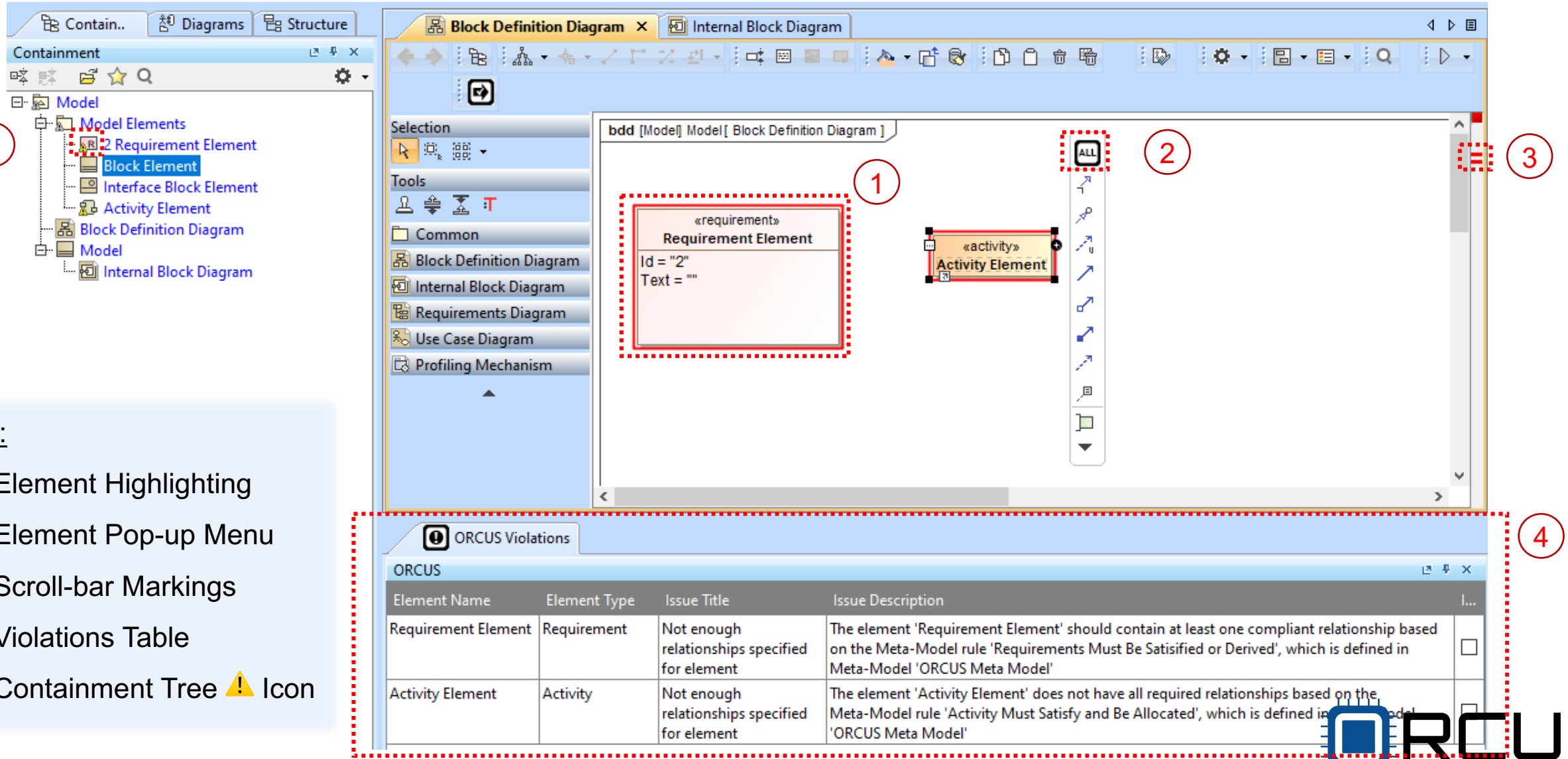


"A logical block must be allocated to physical block."

ORCUS Error Displays – Drag and Drop



ORCUS Error Displays



Key:

(1) Element Highlighting

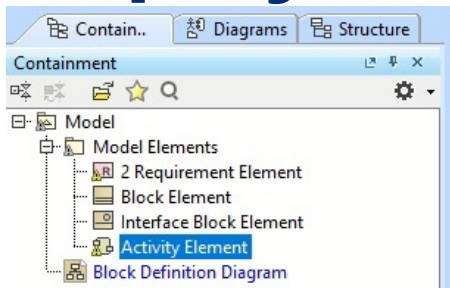
(2) Element Pop-up Menu

(3) Scroll-bar Markings

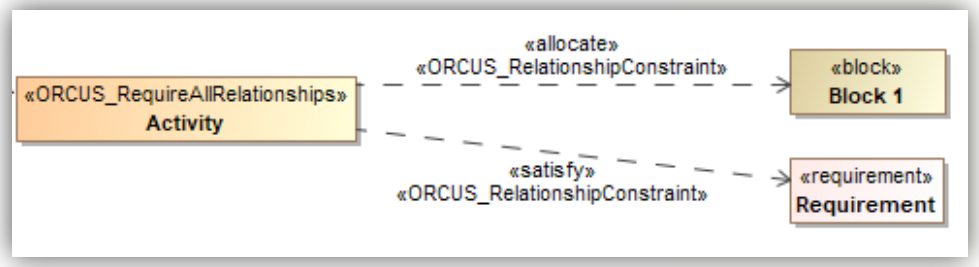
(4) Violations Table

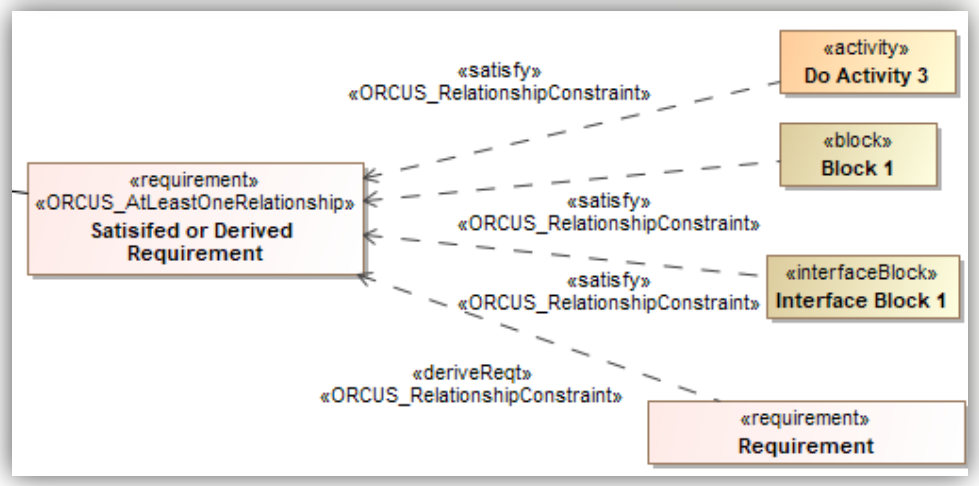
(5) Containment Tree ⚠ Icon

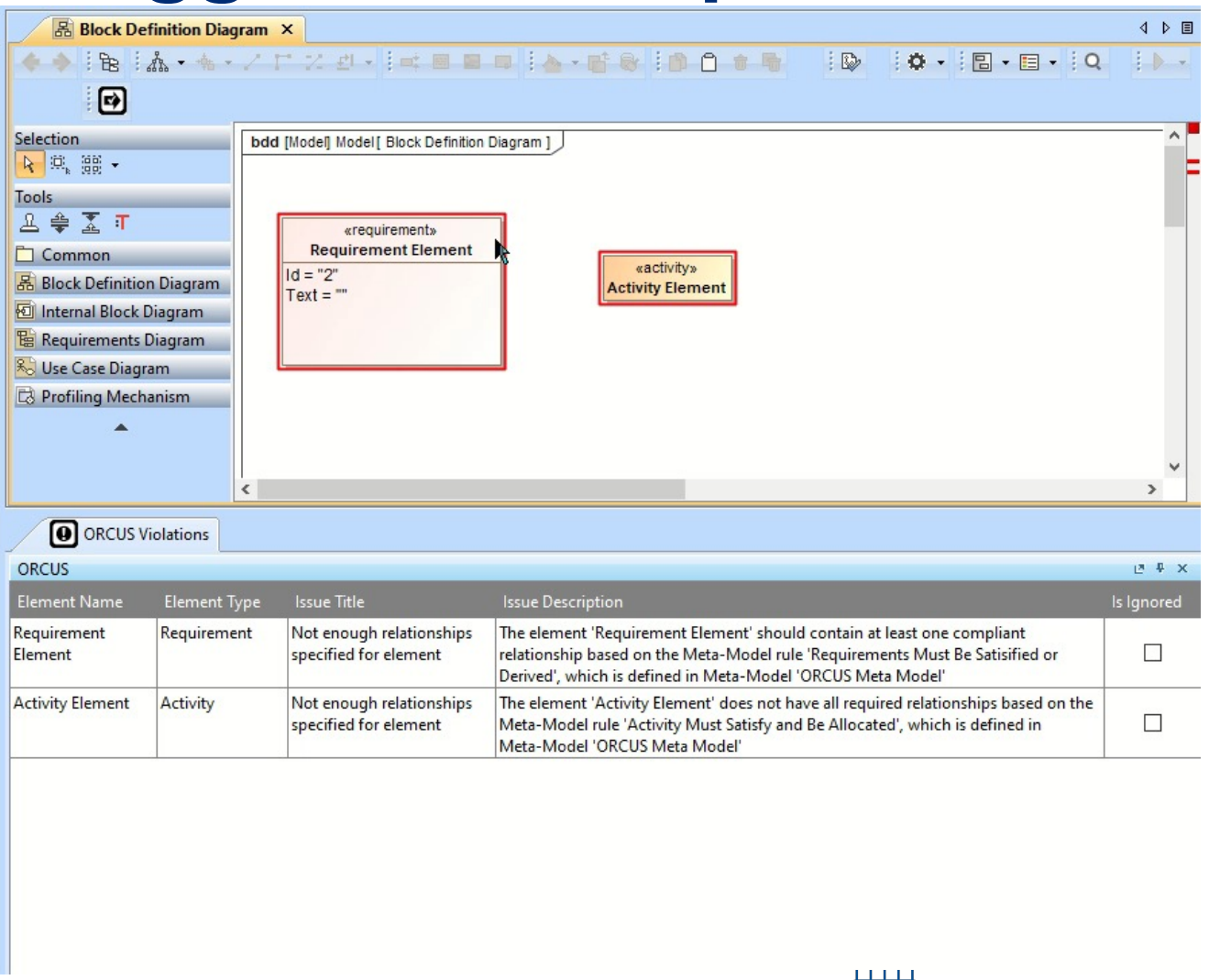
ORCUS Error Displays – Suggested Compliance



Meta-Model Rules

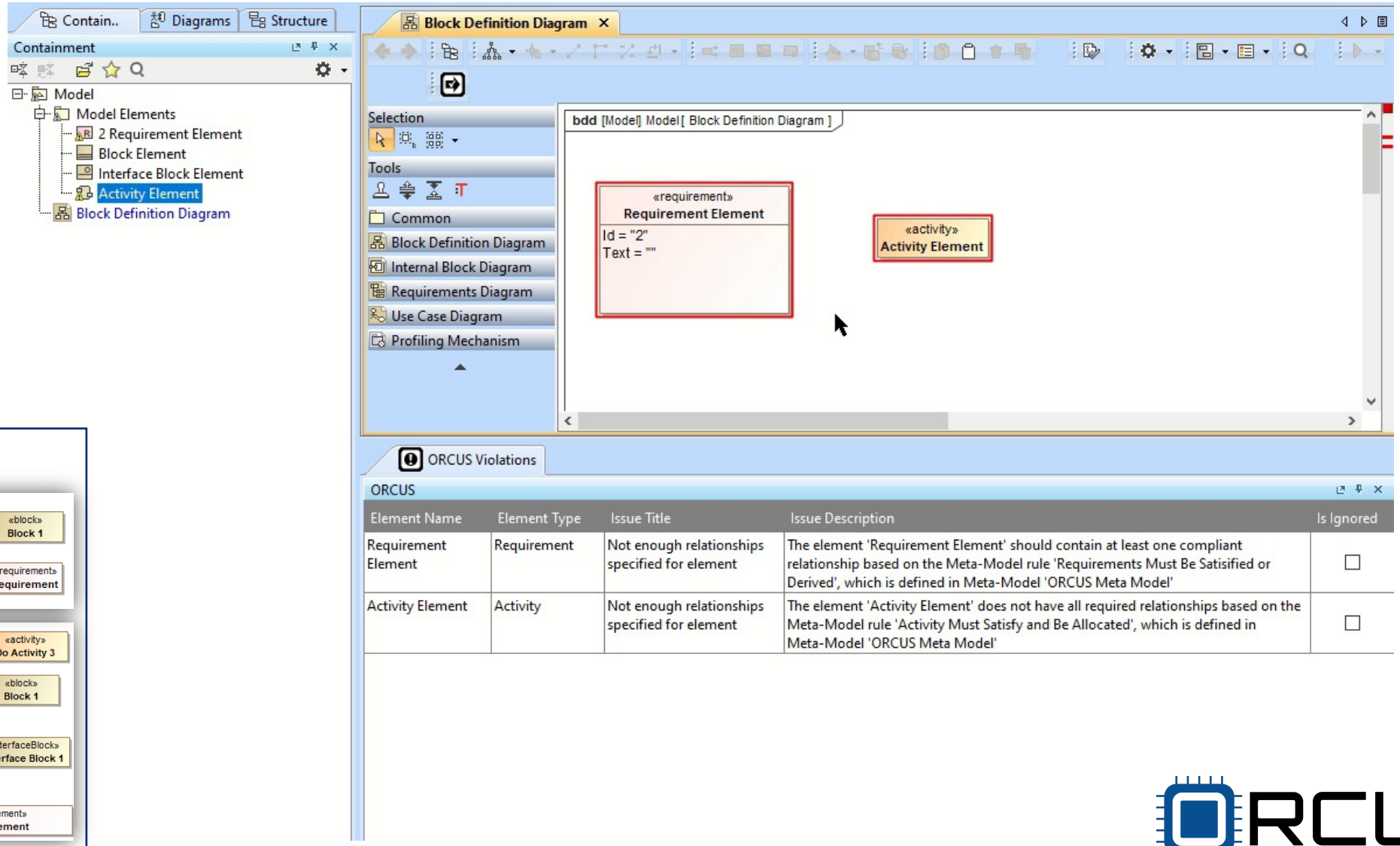




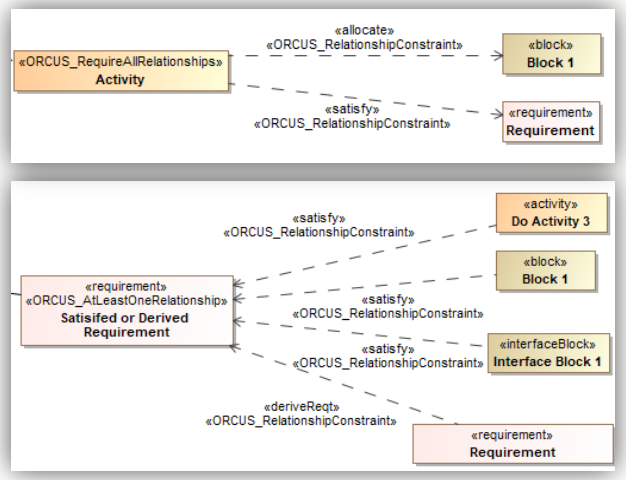


Element Name	Element Type	Issue Title	Issue Description	Is Ignored
Requirement Element	Requirement	Not enough relationships specified for element	The element 'Requirement Element' should contain at least one compliant relationship based on the Meta-Model rule 'Requirements Must Be Satisified or Derived', which is defined in Meta-Model 'ORCUS Meta Model'	☐
Activity Element	Activity	Not enough relationships specified for element	The element 'Activity Element' does not have all required relationships based on the Meta-Model rule 'Activity Must Satisfy and Be Allocated', which is defined in Meta-Model 'ORCUS Meta Model'	☐

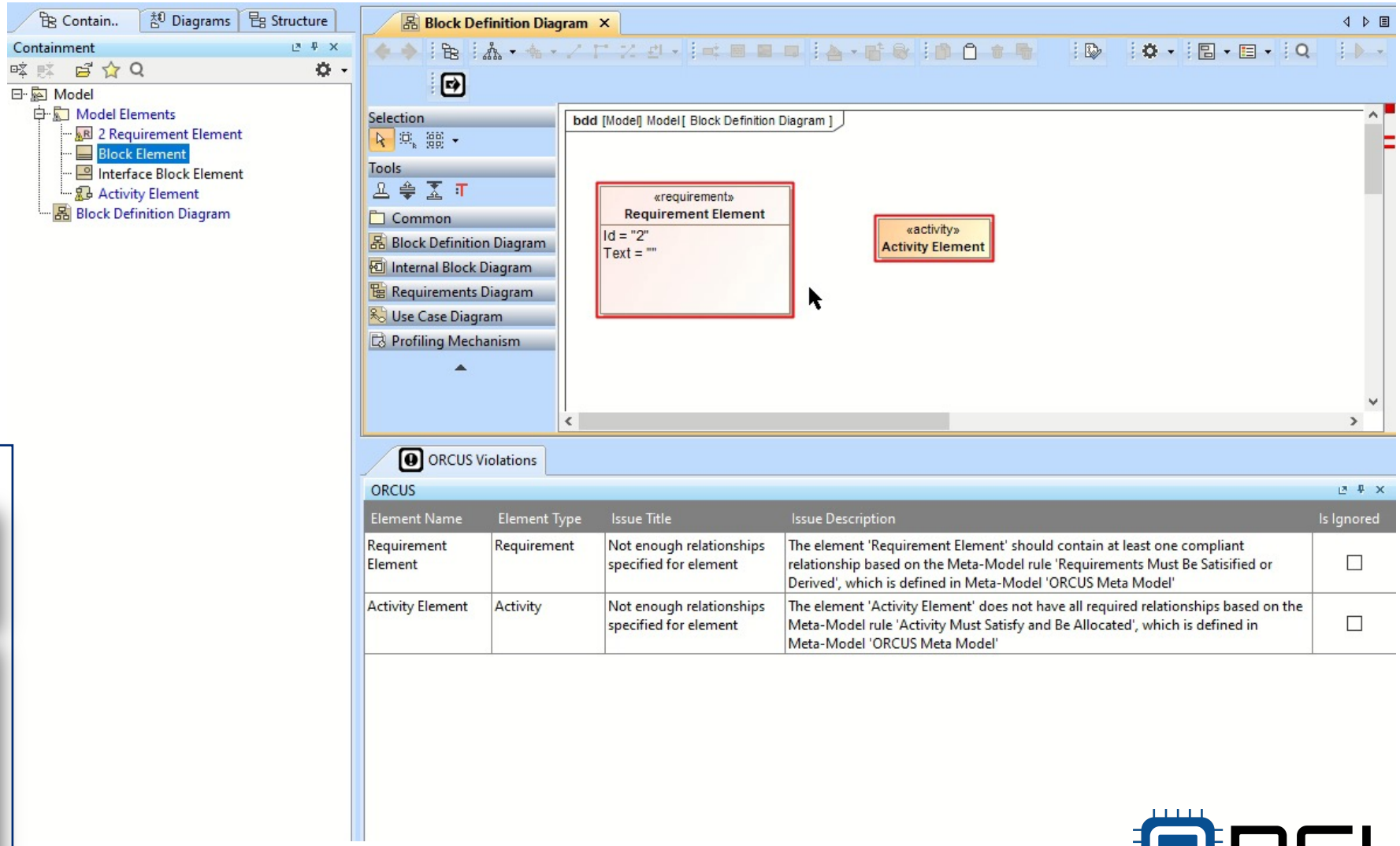
Suggested Compliance – Connect to Element on Diagram



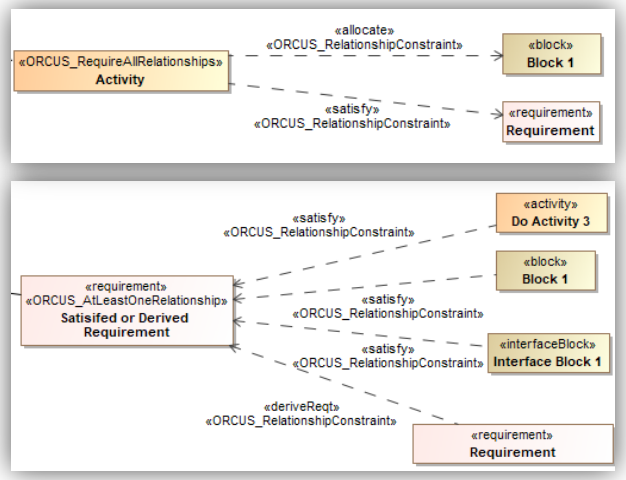
Meta-Model Rules



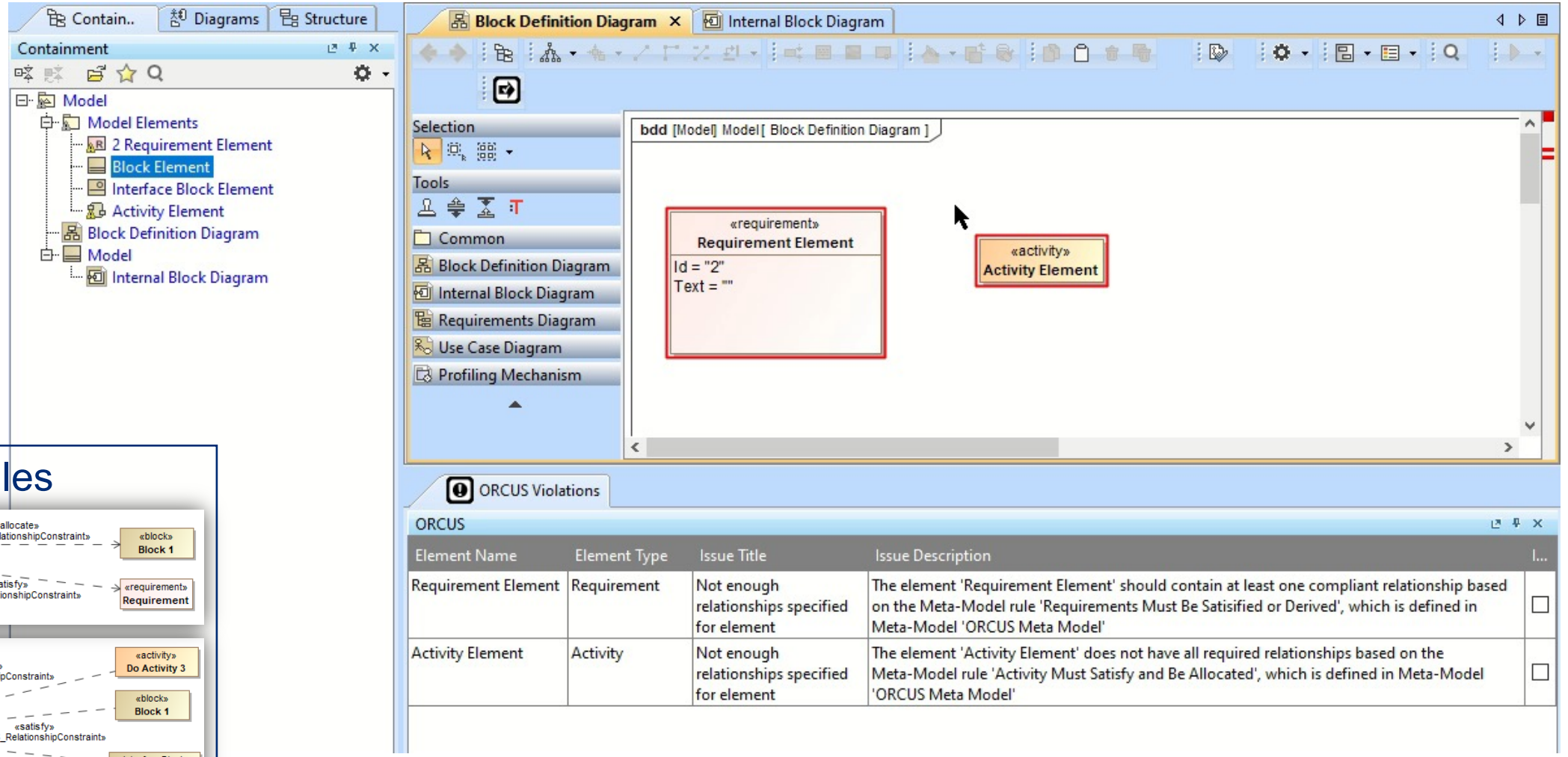
Suggested Compliance – Draw New Element on Diagram



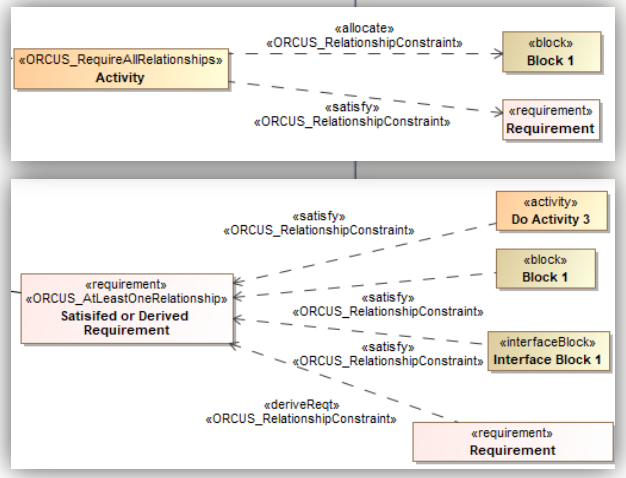
Meta-Model Rules



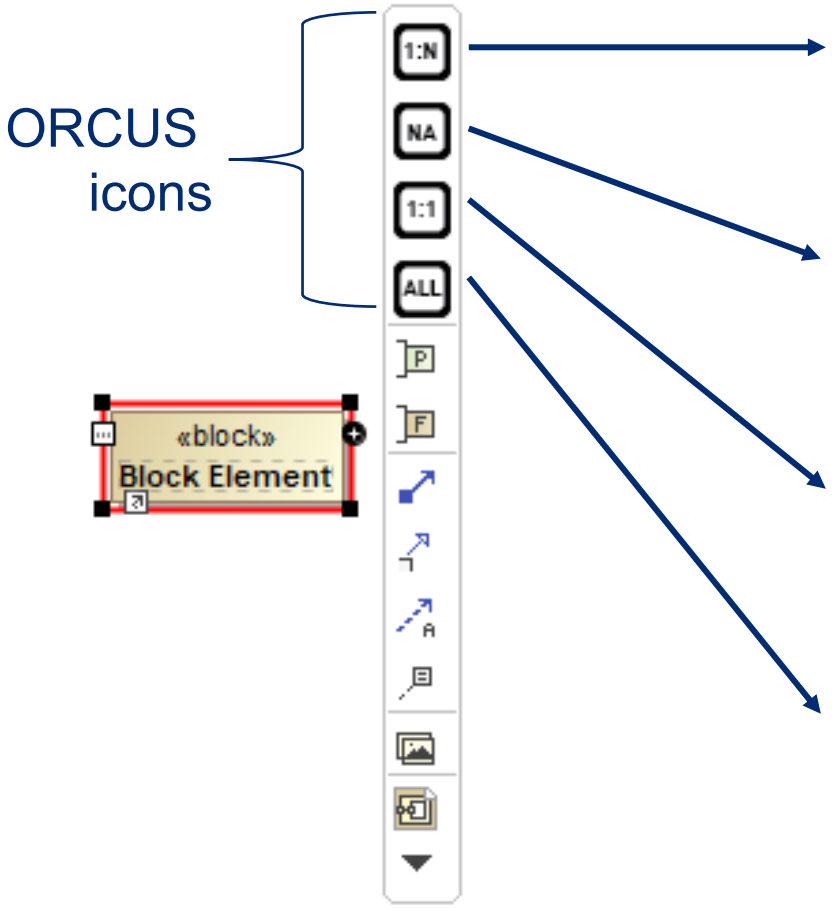
Suggested Compliance – Without Element on Diagram



Meta-Model Rules



Suggested Compliance Icons – Base Rule Types



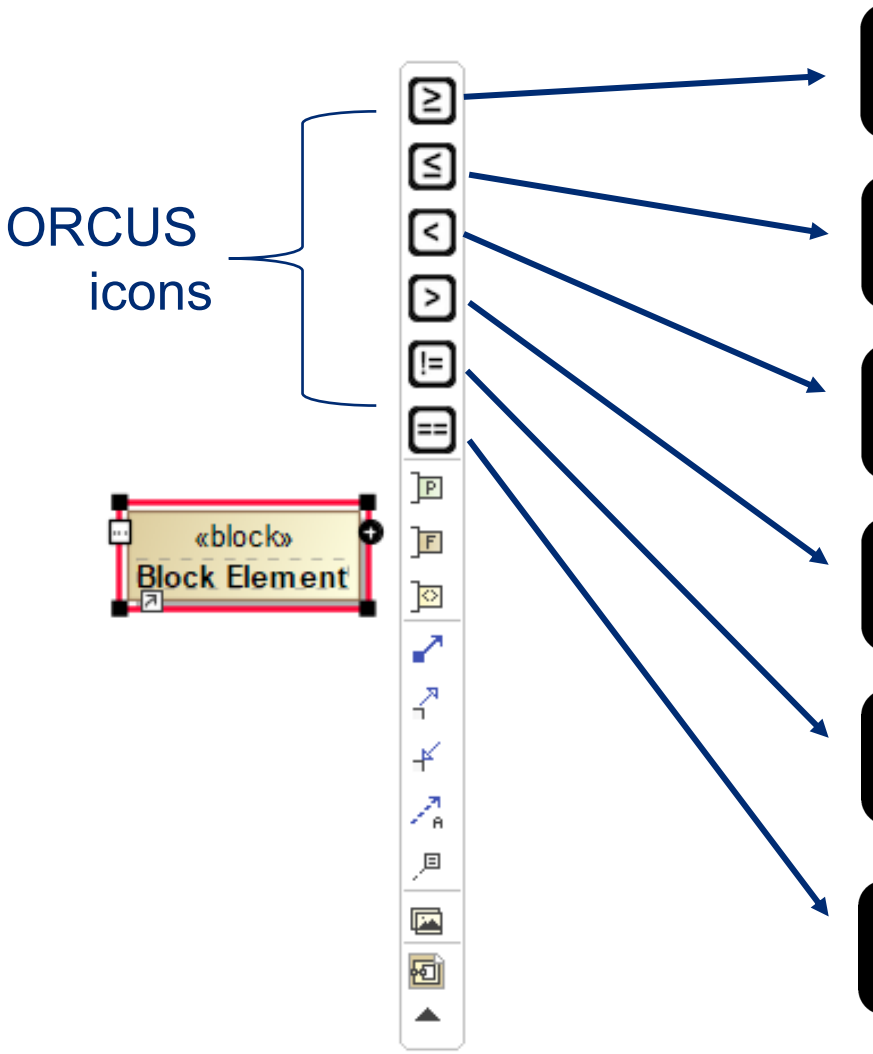
ORCUS_AtLeastOneRelationship
This element is missing a relationship of a relationship set defined for this rule.

ORCUS_ProhibitedRelationshipOnClass
This element currently has a relationship that is expressly prohibited by this rule.

ORCUS_OnlyOneRelationship
This element has multiple occurrences of a relationship set defined for this rule. These relationships must be removed.

ORCUS_RequireAllRelationships
This element is missing a relationship it is required to have.

Suggested Compliance Icons – Custom Rule Types



ORCUS_CustomRule, "[# relationships must be] **≥** [n]"
This element is missing relationships it is expected to have.

ORCUS_CustomRule, "[# relationships must be] **≤** [n]"
This element has excess relationships than what is allowed.

ORCUS_CustomRule, "[# relationships must be] **<** [n]"
This element has excess relationships than what is allowed.

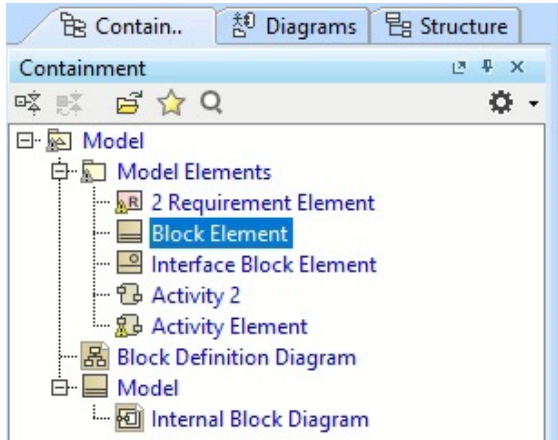
ORCUS_CustomRule, "[# relationships must be] **>** [n]"
This element is missing relationships it is expected to have.

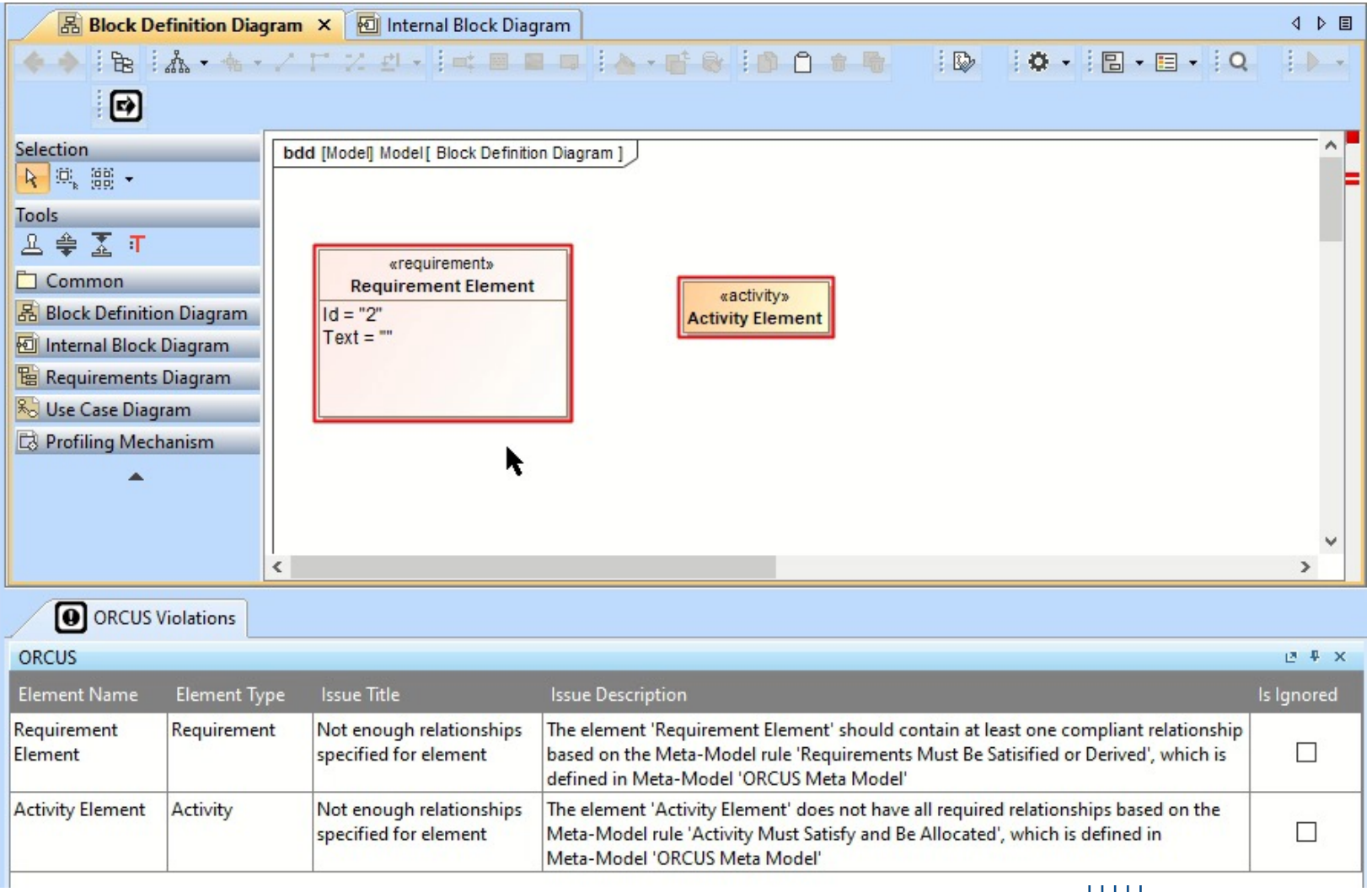
ORCUS_CustomRule, "[# relationships must be] **!=** [n]"
This element has a number of relationships of the defined type that is expressly prohibited.

ORCUS_CustomRule, "[# relationships must be] **==** [n]"
This element has a number of relationships not equal to the defined quantity necessary.

ORCUS

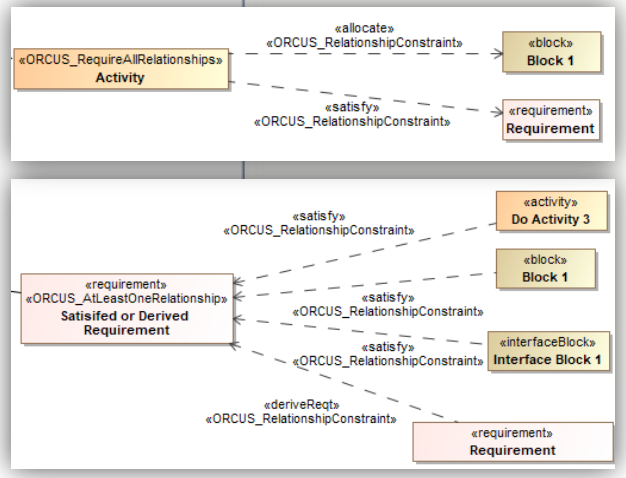
Using ORCUS - Manually Add Relationships



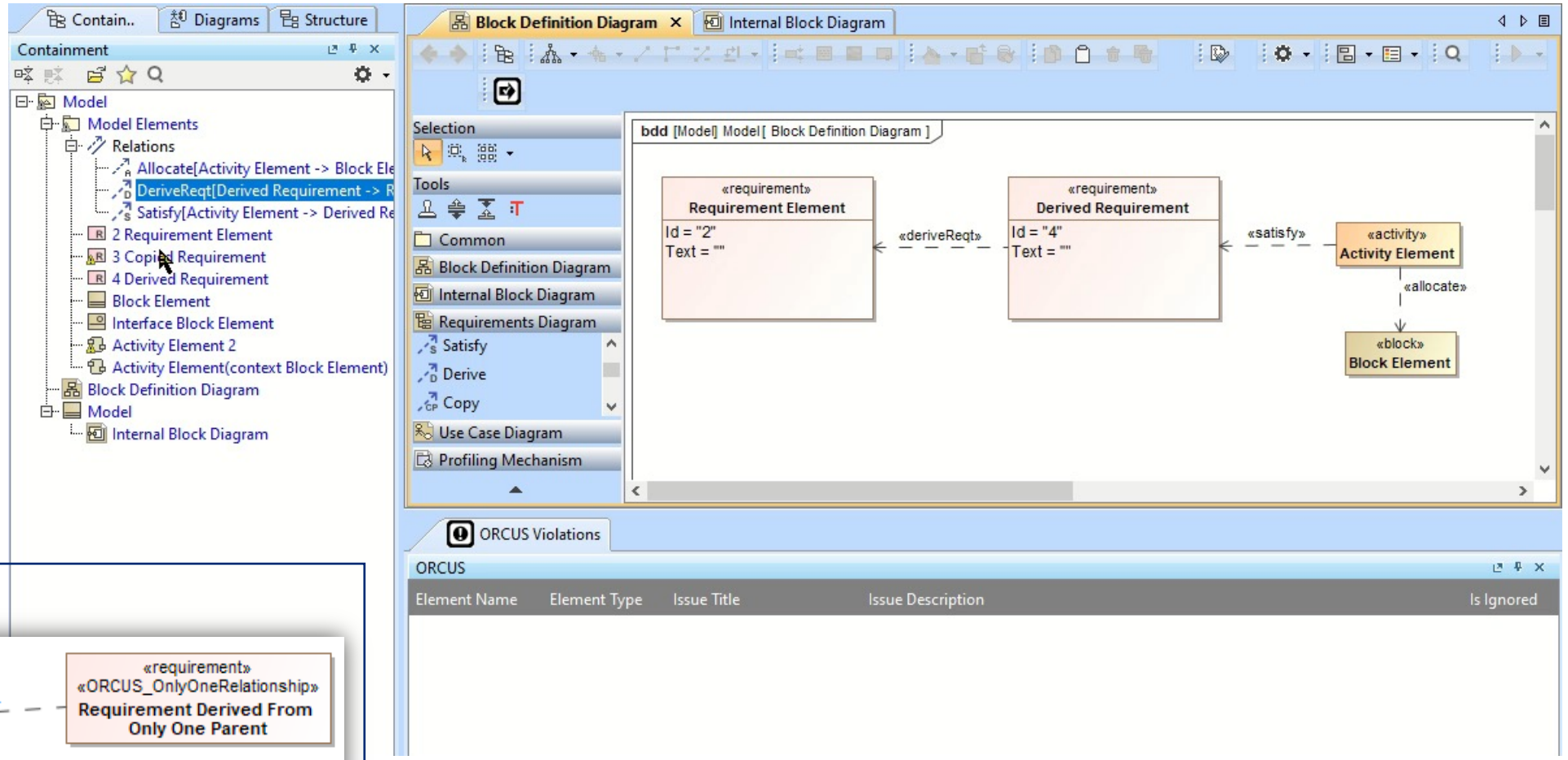


Element Name	Element Type	Issue Title	Issue Description	Is Ignored
Requirement Element	Requirement	Not enough relationships specified for element	The element 'Requirement Element' should contain at least one compliant relationship based on the Meta-Model rule 'Requirements Must Be Satisified or Derived', which is defined in Meta-Model 'ORCUS Meta Model'	☐
Activity Element	Activity	Not enough relationships specified for element	The element 'Activity Element' does not have all required relationships based on the Meta-Model rule 'Activity Must Satisfy and Be Allocated', which is defined in Meta-Model 'ORCUS Meta Model'	☐

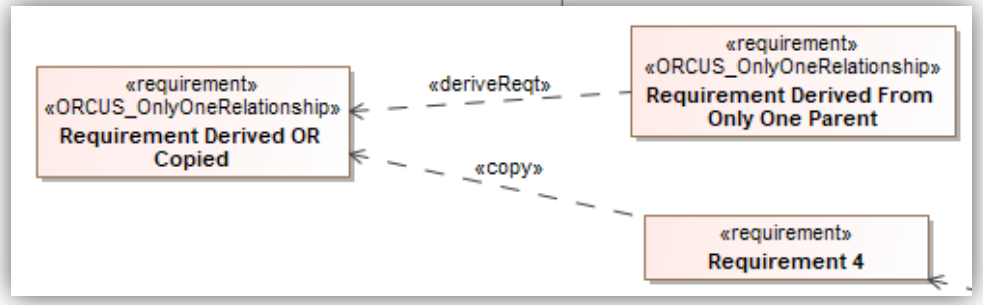
Meta-Model Rules



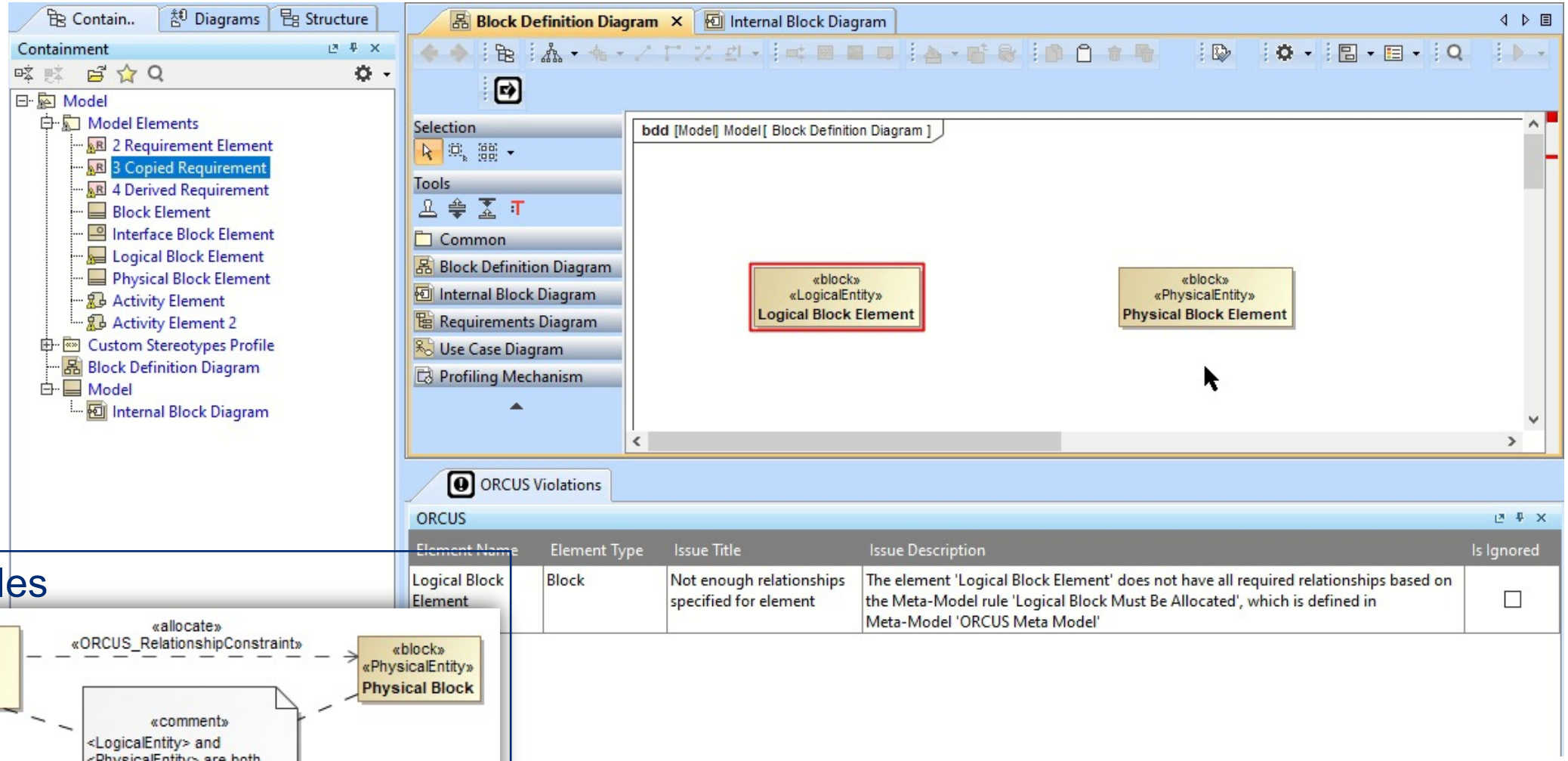
Using ORCUS – Non-Comply with New Violation



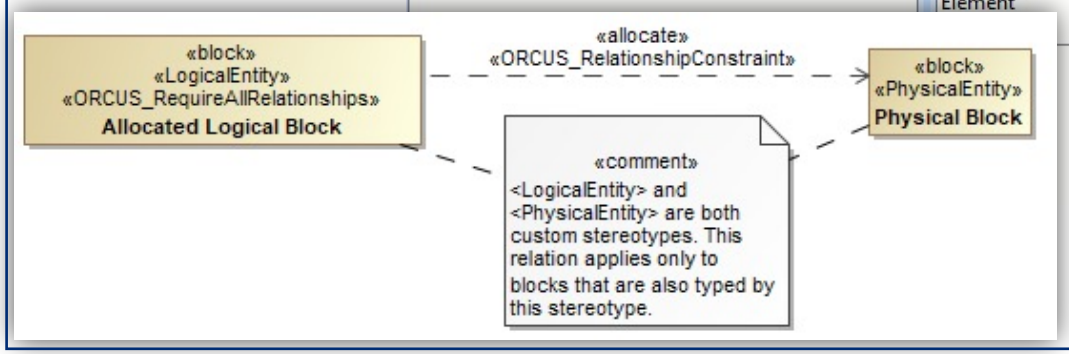
Meta-Model Rules



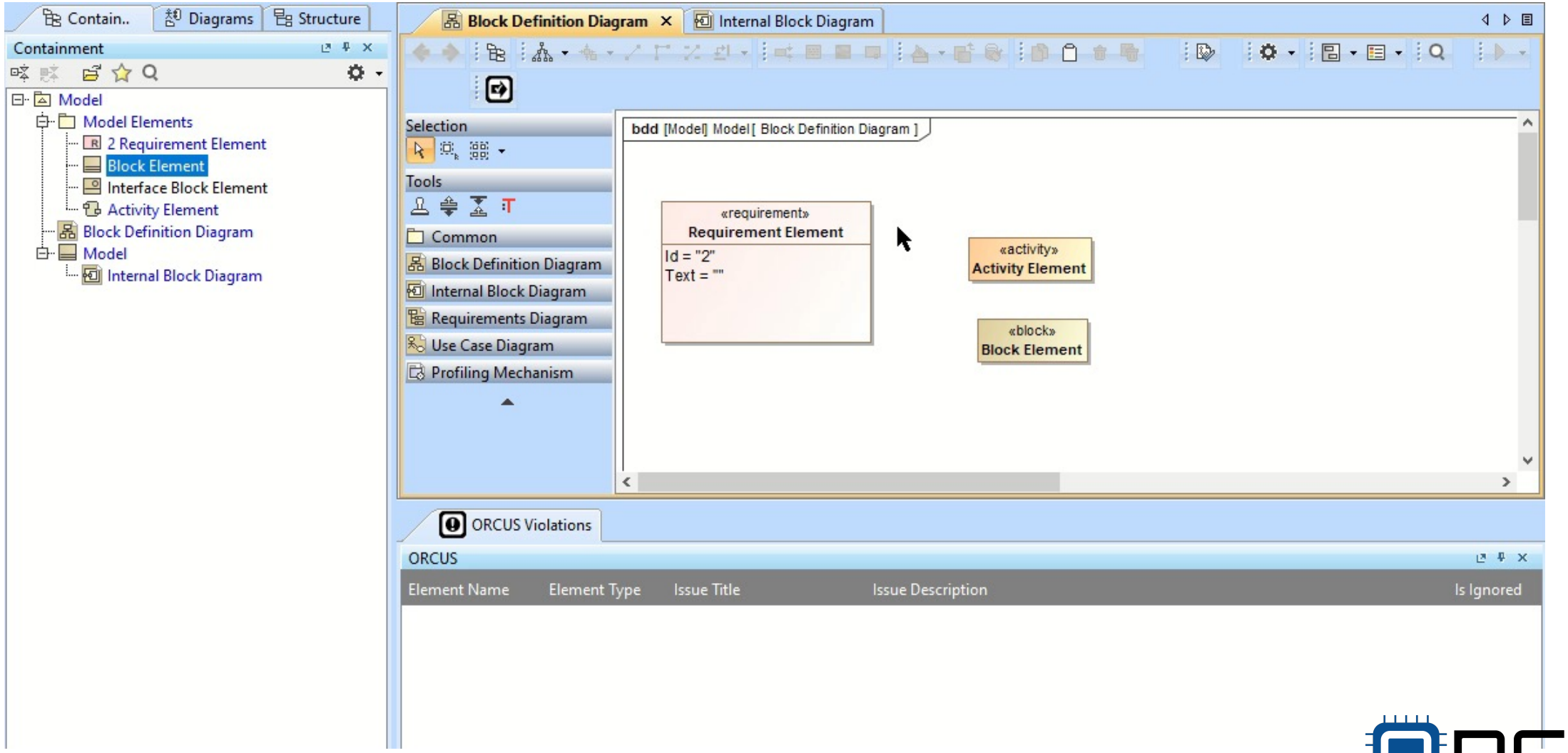
Using ORCUS – Custom Stereotypes



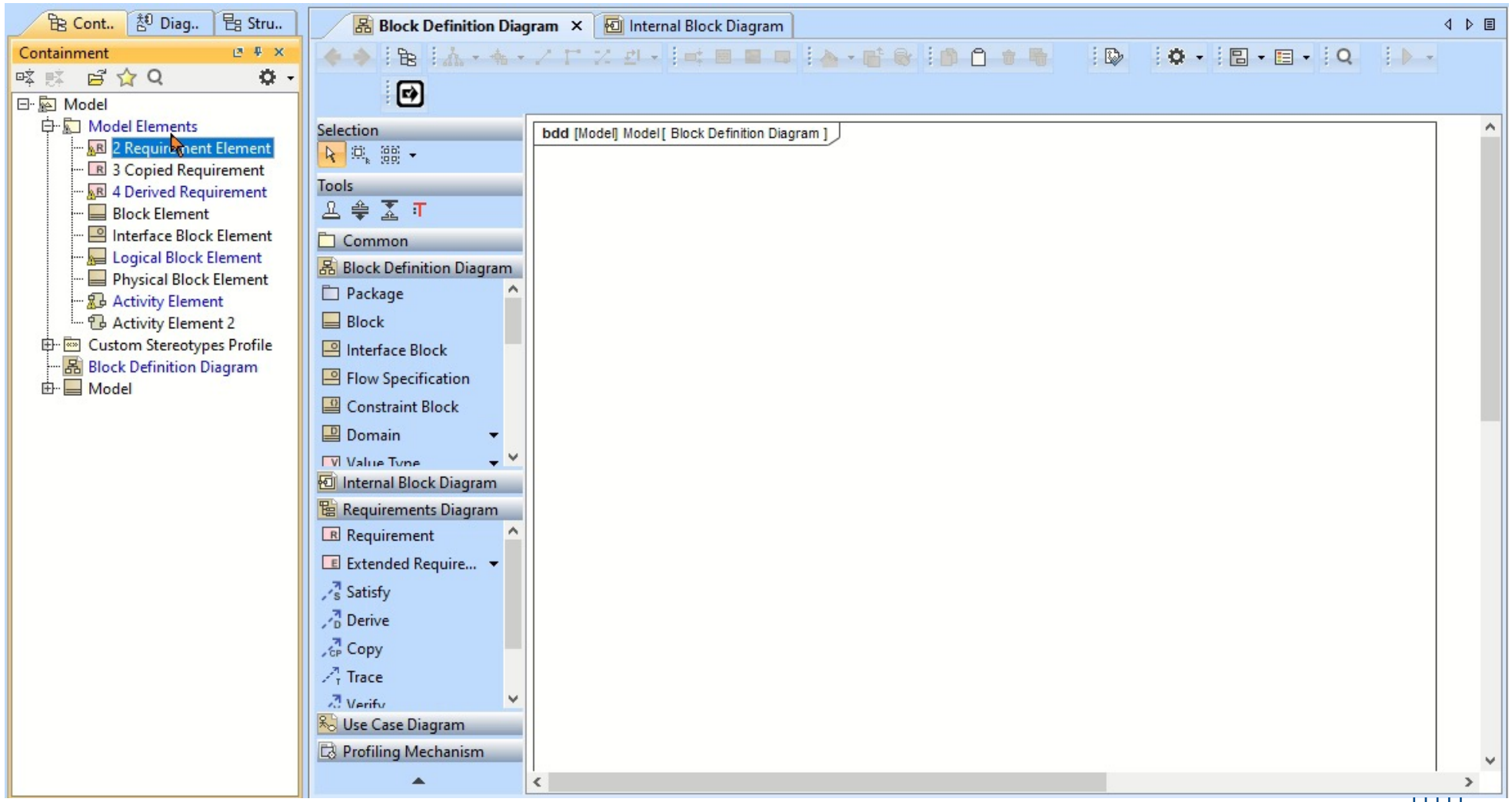
Meta-Model Rules



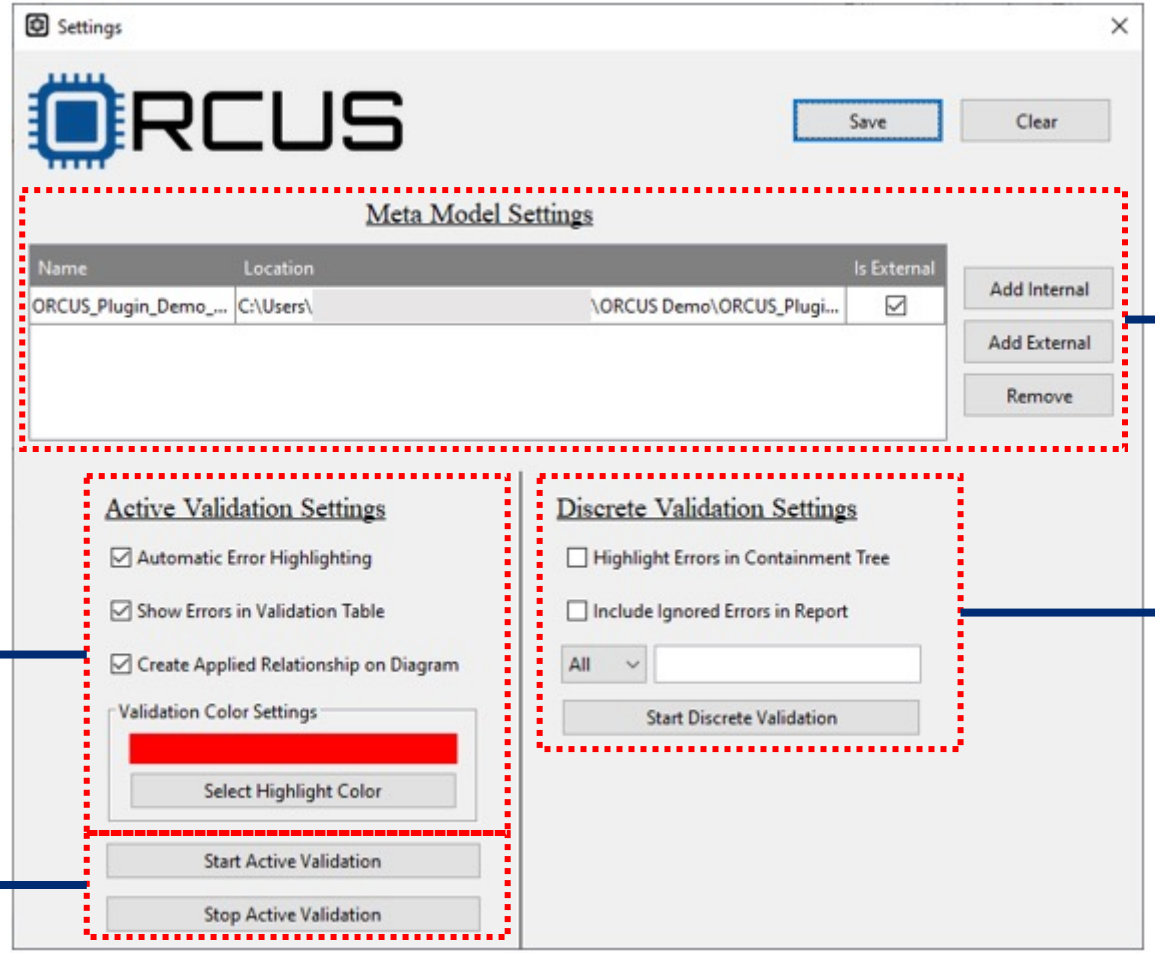
Using ORCUS – Validate Entire Diagram



Using ORCUS – Ease of Linking for Traceability



ORCUS Settings GUI Functions



Load meta-model(s) and view all loaded meta-models (saved into cache).

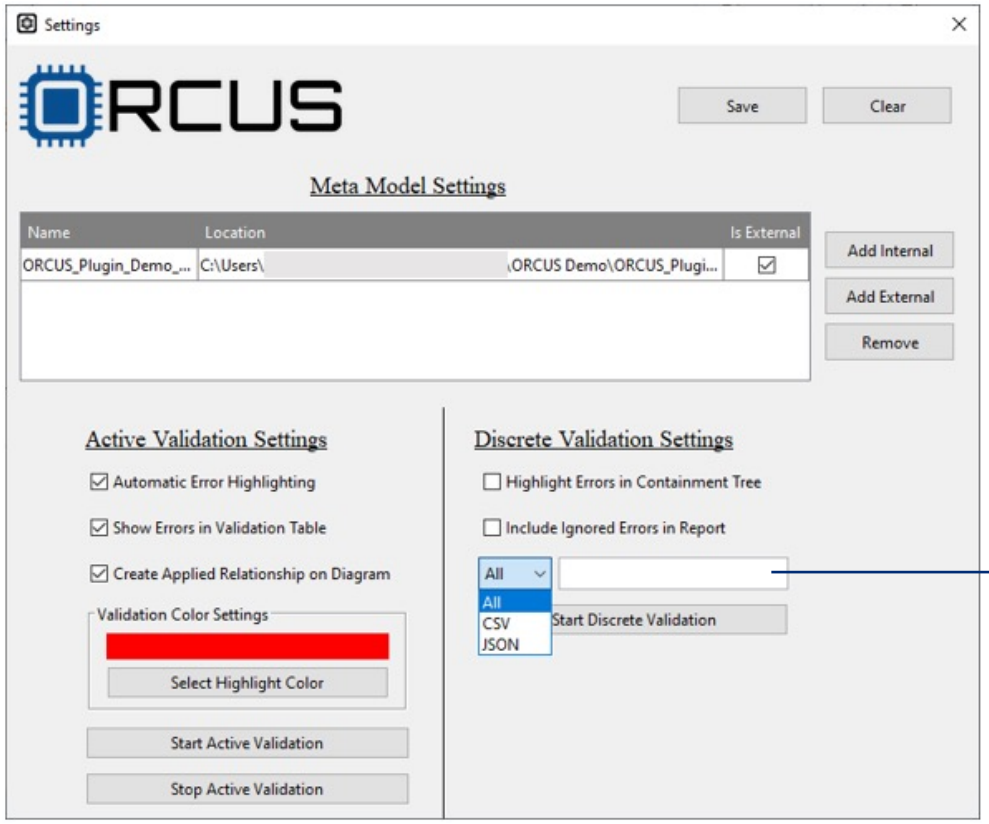
Adjust Discrete Validation display and save settings

Adjust Active Validation display settings

Start/Stop Active Validation (also available in Tools menu)

Using ORCUS – Discrete Validation

- Provides simple method to perform holistic model review or compliance
- Enhances the built-in Cameo validation suite capabilities by using the same rules as the Active Validation



ORCUS StratDDEC Demo.csv
ORCUS StratDDEC Demo.json
All individual meta-model rule violations

ORCUS StratDDEC Demo_Summary.csv
ORCUS StratDDEC Demo_Summary.json
Model compliance metrics

Using ORCUS – CSV export

Model Element Type | **Model Element Name**

Violation Summary

Violation Message

Violation Compliance Actions

Model Element ID	Model Element Server ID	Model Element Type	Model Element Name	Model Element's Owning Model	Violation Rule Name	Violation Rule's Owning Model	Violation Summary	Violation Message	Violation Compliance Actions	ignoreValidationRule
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Requirement	Copied Requirement	Model	Requirements Must Be Satisfied or Derived	ORCUS Meta Model	Not enough relationships specified f	The element 'Copied Requirement' sh	1.) CREATE INCOMING relationship of type <<	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Requirement	Requirement Element 2	Model	Requirements Must Be Satisfied or Derived	ORCUS Meta Model	Not enough relationships specified f	The element 'Requirement Element 2'	1.) CREATE INCOMING relationship of type <<	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Requirement	Derived Requirement	Model	Requirements Must Be Satisfied or Derived	ORCUS Meta Model	Not enough relationships specified f	The element 'Derived Requirement' sl	1.) CREATE INCOMING relationship of type <<	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167485	Activity	Activity Element	Model	Activity Must Satisfy and Be Allocated	ORCUS Meta Model	Not enough relationships specified f	The element 'Activity Element' does n	1.) CREATE OUTGOING relationship of type <<	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167528	Activity	Activity 2	Model	Activity Must Satisfy and Be Allocated	ORCUS Meta Model	Not enough relationships specified f	The element 'Activity 2' does not have	1.) CREATE OUTGOING relationship of type <<	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Allocate	Allocate FROM Block Phy	Model	Activities Must Be Allocated to Blocks	ORCUS Meta Model	Non-compliant Source for relationsh	The relationship type: Allocate has a n	1.) REMOVE relationship of type <<Allocate>>	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Allocate	Allocate FROM Block Phy	Model	Allocate FROM Software TO Hardware	ORCUS Meta Model	Non-compliant Source and Target fo	The relationship type: Allocate has bo	1.) REMOVE relationship of type <<Allocate>>	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Allocate	Allocate FROM Block Phy	Model	Allocate FROM Logical TO Physical	ORCUS Meta Model	Non-compliant Source and Target fo	The relationship type: Allocate has bo	1.) REMOVE relationship of type <<Allocate>>	FALSE
_2021x_2_86e0247_	_2021x_2_86e0247_167525	Allocate	Allocate FROM Block Phy	Model	Activities Must Be Allocated to Blocks	ORCUS Meta Model	Non-compliant Source for relationsh	The relationship type: Allocate has a n	1.) REMOVE relationship of type <<Allocate>>	FALSE

Project ID	Project Name	Project Description	Project Version	Project Last Updated	Project Author	Project File Name	Root Model Package	Total Elements in Project	Total Elements Validated in Project	Total Violations in Project	Rules Defined for Each Meta Model	Number of Violations By Meta Model Rule
_2021x_2_86e0	Demo Model		226			C:\Users\...\ORCUS	Model	74	25	20	Meta Model Name: ORCUS Meta Model	Meta Model Rule Name: Activities Must Be Allocated to Blocks: 3
247_167534273						Demo\Demo					--Interface Blocks Can Satisfy Requirements	Meta Model Rule Name: Allocate FROM Software TO Hardware: 3
8385_674529_2						Model.mdzip					--Single Parent Requirement	Meta Model Rule Name: Logical Block Must Be Allocated: 1
795											--Association of Subsystem and Logical	Meta Model Rule Name: Activity Must Satisfy and Be Allocated: 4
											--Logical Block Must Be Allocated	Meta Model Rule Name: Requirements Must Be Satisfied or Derived: 6
											--Activity Must Satisfy and Be Allocated	Meta Model Rule Name: Allocate FROM Logical TO Physical: 3
											--Association of Component and Physical	
											--Activities Must Be Allocated to Blocks	
											--Only Verify Requirements from Test Case	
											--Requirement Can Only Derive OR Copy Once	
											--Requirements Can Derive Into Requirements	
											--Test Case Must Verify Requirement	
											--Blocks Cannot Trace	
											--Blocks Can Satisfy Requirements	
											--Subsystems Composed of Components	
											--Activity Refines Single Use Case	
											--Components Associated with Physical Elements	
											--Subsystems Associated with Logical Elements	
											--Association of System and Logical	
											--Allocate FROM Logical TO Physical	
											--Activities Can Satisfy Requirements	
											--Prohibit Abstractions	
											--Software Must Be Allocated	
											--Systems Are Composed of Subsystems	
											--Allocate FROM Software TO Hardware	
											--Systems Associated with Logical Elements	
											--Activities Satisfy Requirements	

Total Violations in Project | **Rules Defined for Each Meta Model**

Number of Violations By Meta Model Rule

Using ORCUS – JSON export


```
[ {
  "Model Element ID" : "_2021x_2_86e0247_1675292327786_755194_3060",
  "Model Element Server ID" : "_2021x_2_86e0247_1675292327786_755194_3060",
  "Model Element Type" : "Requirement",
  "Model Element Name" : "Copied Requirement",
  "Model Element's Owning Model" : "Model",
  "Violation Rule Name" : "Requirements Must Be Satisified or Derived",
  "Violation Rule's Owning Model" : "ORCUS Meta Model",
  "Violation Summary" : "Not enough relationships specified for element",
  "Violation Message" : "The element 'Copied Requirement' should contain at least one compliant relation
  "complianceOptions" : [ {
    "complianceOptionAction" : "CREATE",
    "relationshipDirection" : "INCOMING",
    "relationshipType" : "DeriveReqt",
    "relationshipStereotypes" : "",
    "relationshipSourceType" : "Requirement"
    "relationshipSourceName" : "",
    "relationshipSourceStereotypes" : "Requi
    "relationshipTargetType" : "",
    "relationshipTargetName" : "Copied Requi
    "relationshipTargetStereotypes" : "",
    "isDirectedRelationship" : false
  }, {
    "complianceOptionAction" : "CREATE",
    "relationshipDirection" : "INCOMING",
    "relationshipType" : "Satisfy",
    "relationshipStereotypes" : "",
    "relationshipSourceType" : "Block",
    "relationshipSourceName" : "",
    "relationshipSourceStereotypes" : "Block
    "relationshipTargetType" : "",
    "relationshipTargetName" : "Copied Requi
    "relationshipTargetStereotypes" : "",
    "isDirectedRelationship" : false
  }, {
```

```
{
  "Project ID" : "_2021x_2_86e0247_1675342738385_674529_2795",
  "Project Name" : "Demo Model",
  "Project Description" : "",
  "Project Version" : "226",
  "Project Last Updated" : null,
  "Project Author" : null,
  "Project File Name" : "C:\\Users\                    \\ORCUS Demo\\Demo Model.mdzip",
  "Root Model Package" : "Model",
  "Total Elements in Project" : 74,
  "Total Elements Validated in Project" : 25,
  "Total Violations in Project" : 20,
  "Rules Defined for Each Meta Model" : "Meta Model Name: ORCUS Meta Model\n--Interface Blocks Can Satisfy Requ
  "Number of Violations By Meta Model Rule" : "Meta Model Rule Name: Activities Must Be Allocated to Blocks: 3\
  "totalRulesByMetaModel" : {
    "ORCUS Meta Model" : [ "Interface Blocks Can Satisfy Requirements", "Single Parent Requirement", "Associati
  },
  "totalViolationsByRule" : {
    "Activities Must Be Allocated to Blocks" : 3,
    "Allocate FROM Software TO Hardware" : 3,
    "Logical Block Must Be Allocated" : 1,
    "Activity Must Satisfy and Be Allocated" : 4,
    "Requirements Must Be Satisified or Derived" : 6,
    "Allocate FROM Logical TO Physical" : 3
  }
```

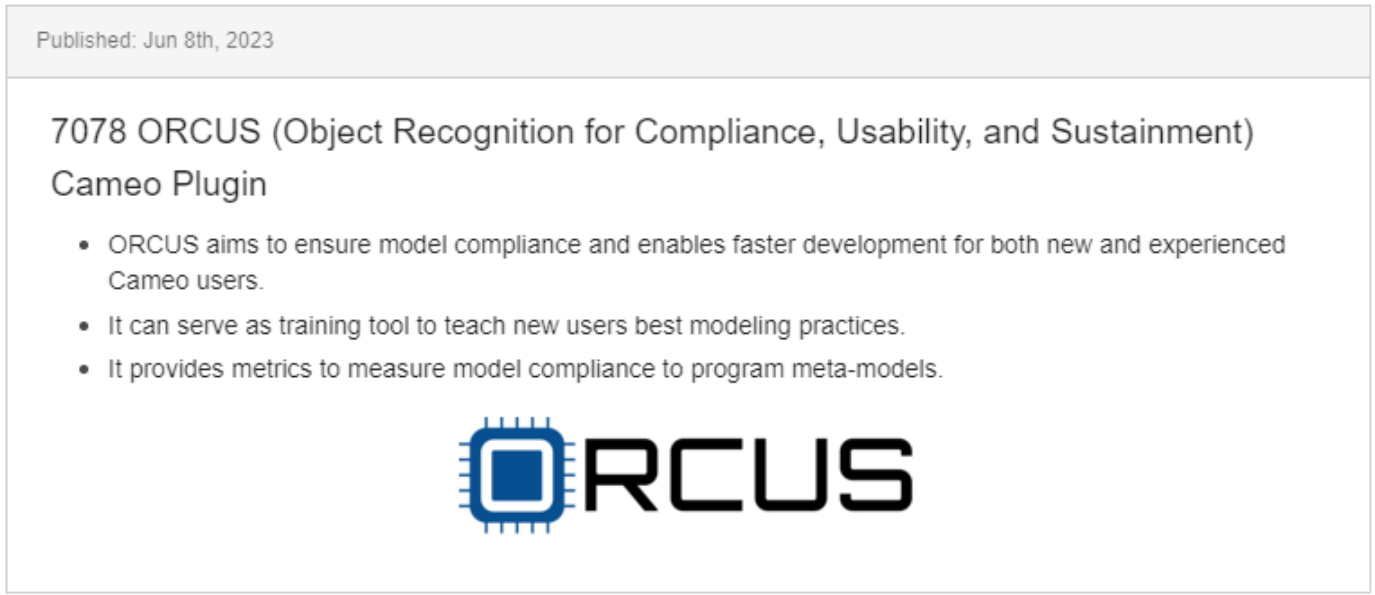
Summary of ORCUS

- JHUAPL-developed plug-in for Cameo 2021x to enable a(n):
  - Simpler method for ensuring compliance to meta-models
  - Easier method for gauging metrics on model compliance
  - Faster teaching of modeling patterns, practices, and standards to newer Modelers
  - Interactive user experience within Cameo for adhering to meta-models
- ORCUS was designed with the flexibility to simply convert existing meta-models and to be able to validate against multiple meta-models simultaneously
- ORCUS has two compliance validation modes:
  - **Active Validation** for working in the model
  - **Discrete Validation** for exporting model metrics

Object Recognition for Compliance, Usability, and Sustainment

Available for Download!

- ORCUS is available for public download:
  - https://www.jhuapl.edu/techtransfer/Technologies/Licensing
- Please email orcus@jhuapl.edu with any questions, feedback or enhancement requests!



JOHNS HOPKINS

APPLIED PHYSICS LABORATORY