

# Digital Modeling Training: Modeling Requirements Session 2

26 July 2023

Nataliya Shevchenko

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

**NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.**

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

DM23-0647

# Digital Modeling Training Outline

Section 1: Definitions and Types

Section 2: Categories and Organization

Section 3: Traceability and Relationships

Section 4: Diagrams tools

Section 5: Demo in Cameo

Digital Modeling Training

# Section 1: Definitions and Types



# Requirements in SE: Definition

An initial description of a problem that the future system should solve.

Requirements can describe **logical** and **physical** aspects of the future system.

Comes in the form of:

- Document with users' wishes
- Outcomes of business-analysis processes
- Change request
- Verbal request during a user forum
- Request from sponsors

# Requirements in SE: Types

**Business requirements:** High-level statements of the goals, objectives, or needs of an organization. They usually describe opportunities or problems that pertain to the organization.

**User requirements:** Mid-level statements of the needs of a particular stakeholder or group of stakeholders. They usually describe how someone wants to interact with the intended solution. User requirements are often situated midway between the high-level business requirements and more detailed solution requirements.

# Requirements in SE: Types

**System requirements:** Usually detailed statements of capabilities, behavior, and information that the solution will need including detailed statements of the conditions under which the solution must remain effective, qualities that the solution must have, or constraints within which it must operate.

- Functional requirements
- Non-functional requirements - often called quality attributes or “ilities”:
  - Security
  - Usability
  - Testability
  - Modifiability
  - ...

# MBSE

**Model-based systems engineering is a formalized methodology that supports the requirements, design, analysis, verification, and validation associated with the development of complex systems.**

**Modeling Language:** SySML

**Modeling Environment:** Digital modeling environment

# Requirements in SysML: Types

- Requirements
- Business requirement
- Usability requirement
- Functional requirement
- Performance requirement
- Interface requirement
- Physical requirement
- Design constraint

# Requirements Mapping: SE to SysML

- Business requirement → SysML *business requirement*
- User requirement → SysML *generic requirement* with *user requirement* stereotype
- System functional requirement → SysML *generic requirement* with *system requirement* stereotype or SysML *functional requirement* subclass
- System non-functional requirements → SysML *design constraint*, *usability requirement*, *performance requirement*, *interface requirement*, *physical requirement*

Digital Modeling Training

## Section 2: Categories and Organization



# Organization

In SysML, the main method of organization is packages.

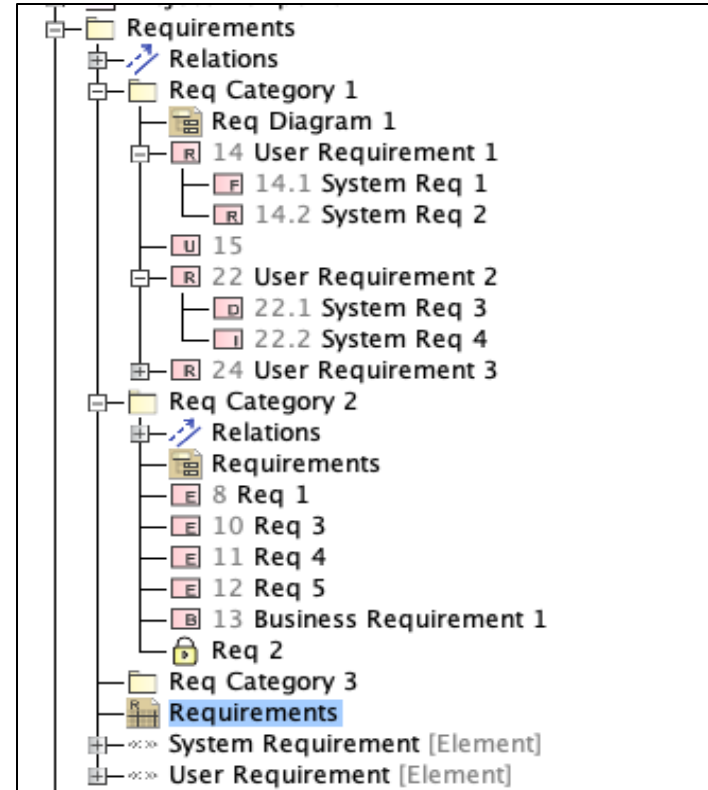
**Packages** are folder-like structures that contain requirements in a hierarchical manner.

They can have their own internal structure to manage the complexity of the requirements.

# Categories as Initial Analysis

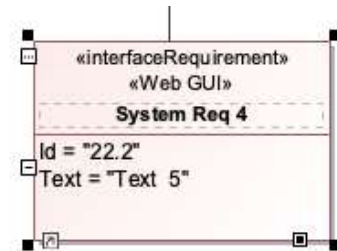
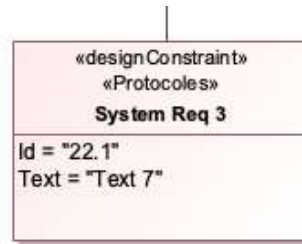
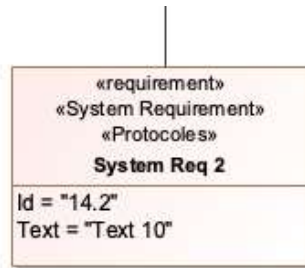
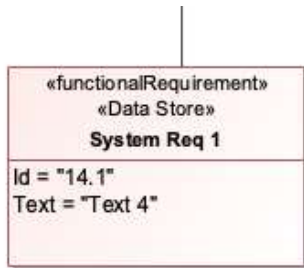
**Categorization** is a system engineering techniques for requirements analysis.









- Categories based on:
  - Type of requirements
  - Functionality
  - Part of a business process
  - Subsystems
  - Components
- Categories as Packages



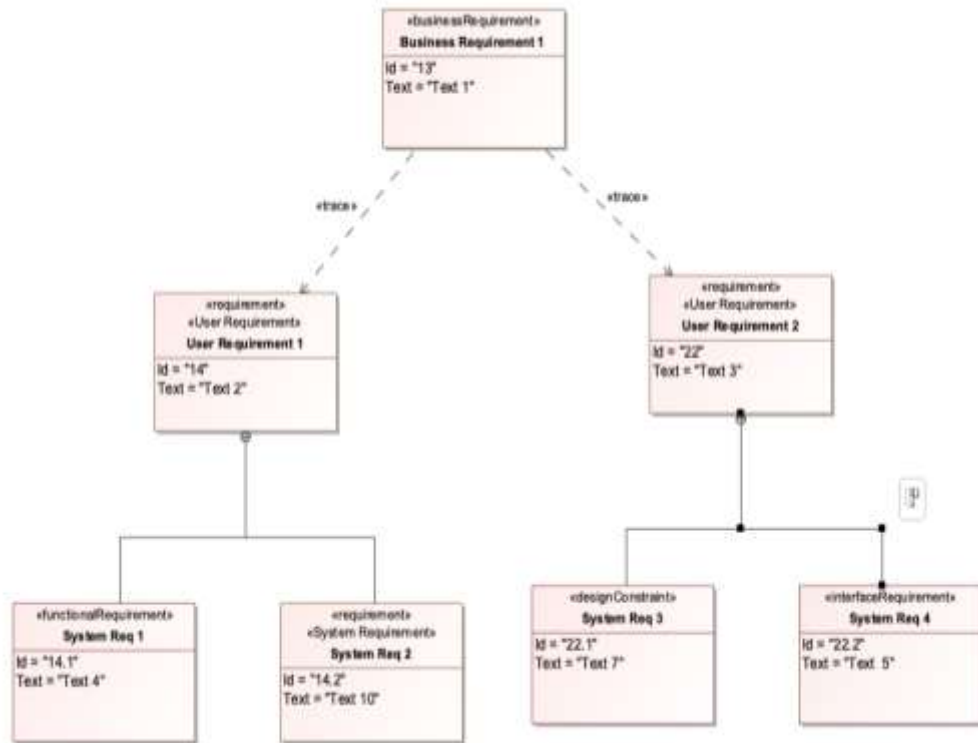
# Custom Stereotypes

## Categories as Custom Stereotypes



#	Name	Text	Applied Stereotype
1	 14.1 System Req 1	Text 4	 functionalRequirement [Class] «>> Data Store [Element]
2	 14.2 System Req 2	Text 10	 Requirement [Class] «>> System Requirement [Element] «>> Protocoles [Element]
3	 22.1 System Req 3	Text 7	 designConstraint [Class] «>> Protocoles [Element]
4	 22.2 System Req 4	Text 5	 interfaceRequirement [Class] «>> Web GUI [Element]

# Decomposition



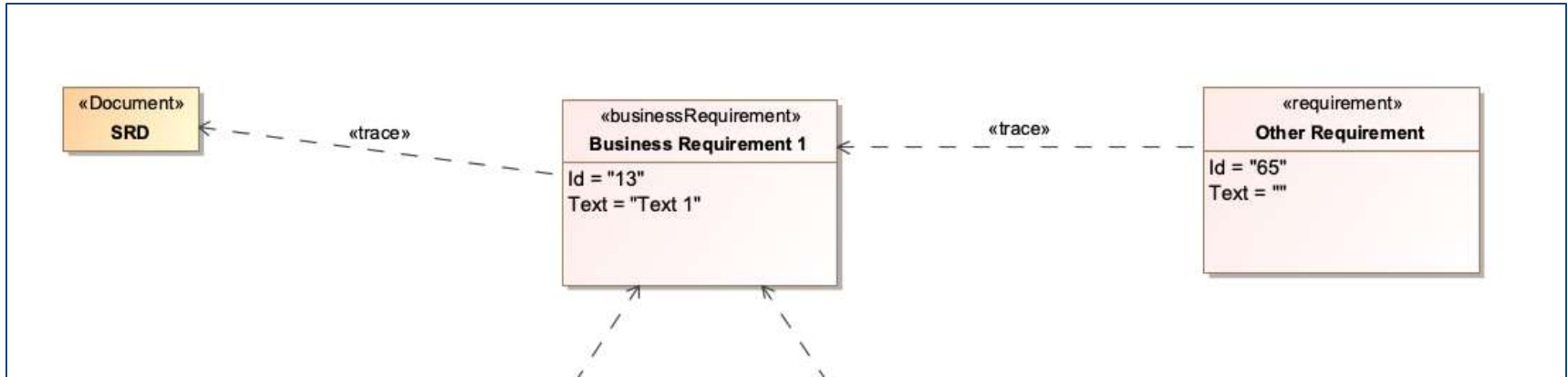
Digital Modeling Training

## Section 3: Traceability and Relationships



# Relationship: *Trace*

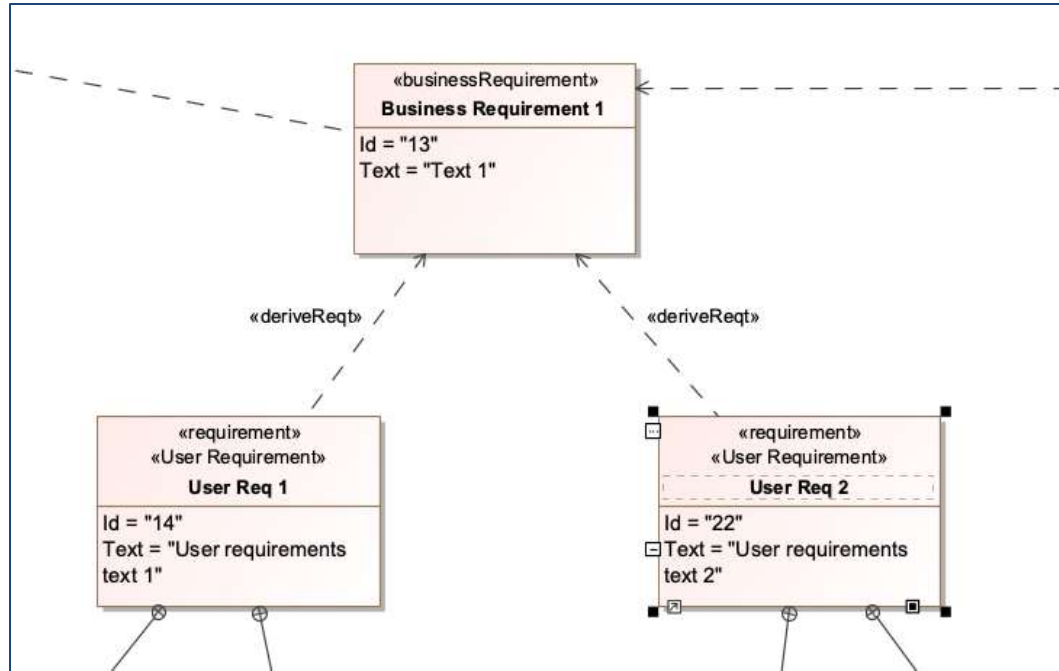
- Requirement was published in the System Requirements Document (SRD)
- Requirement can be traced to another requirement (weakest relationship)



# Relationship: *Derive*

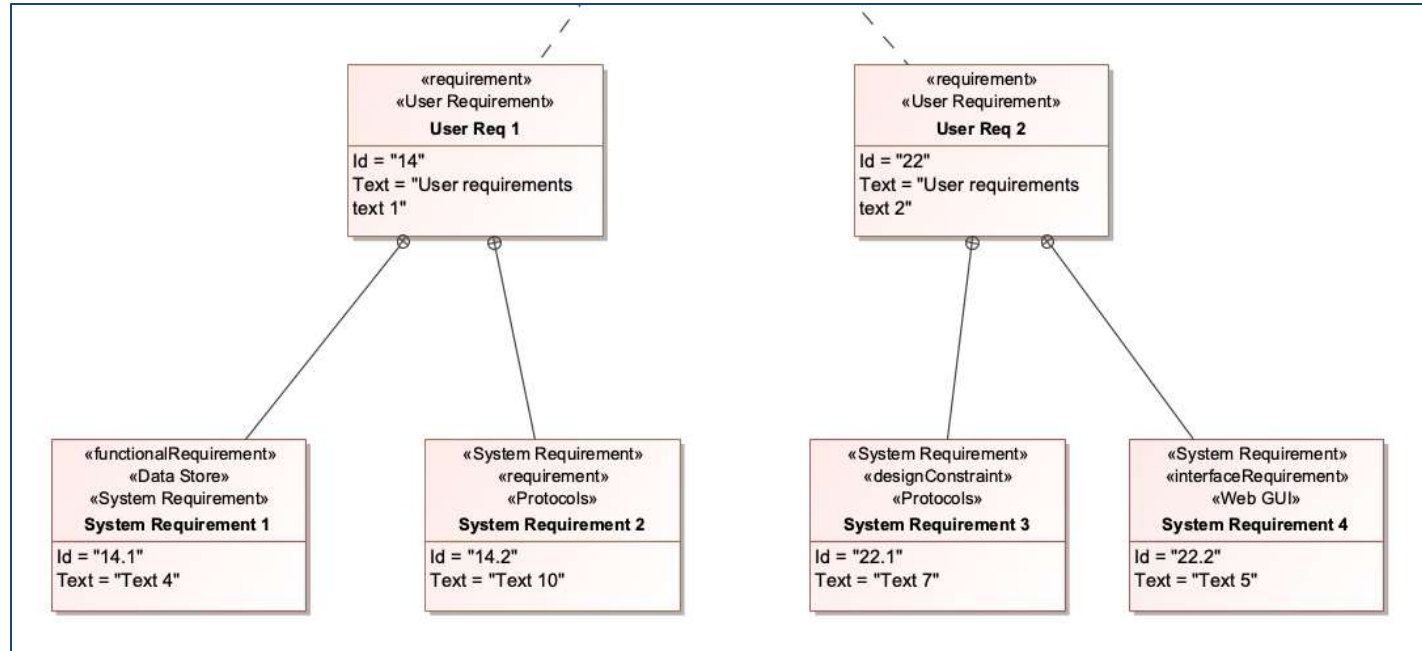
Other requirements were derived from this requirement:

- Parent-child
- Interpretation
- Level of abstraction



# Relationship: *Containment*

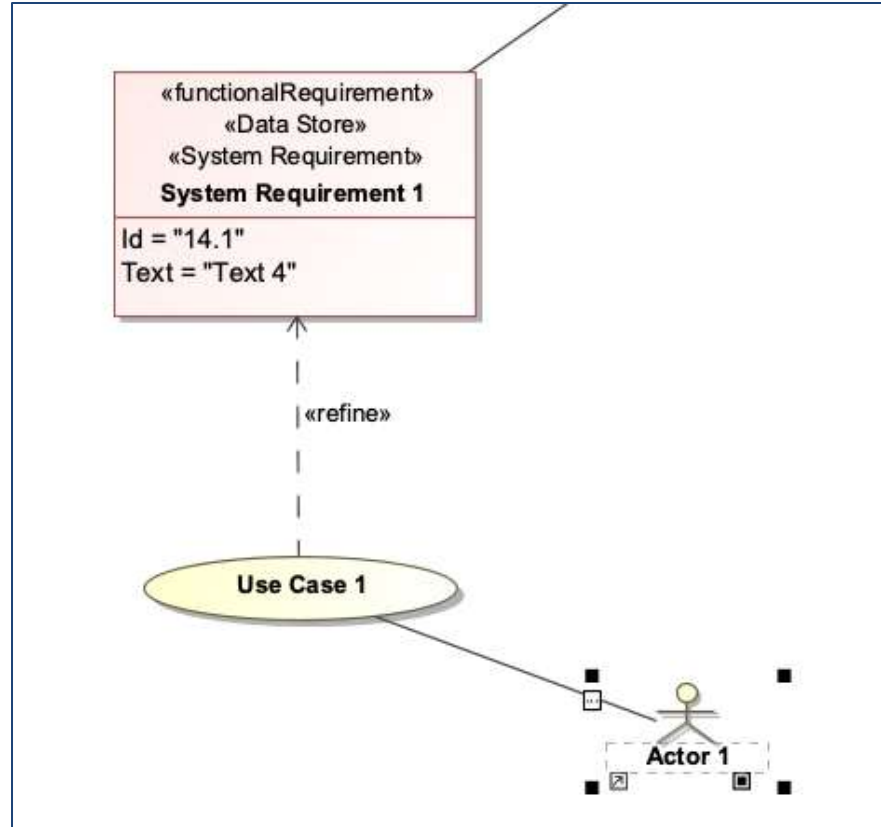
Requirement was decomposed



# Relationship: *Refine*

Requirement is refined by:

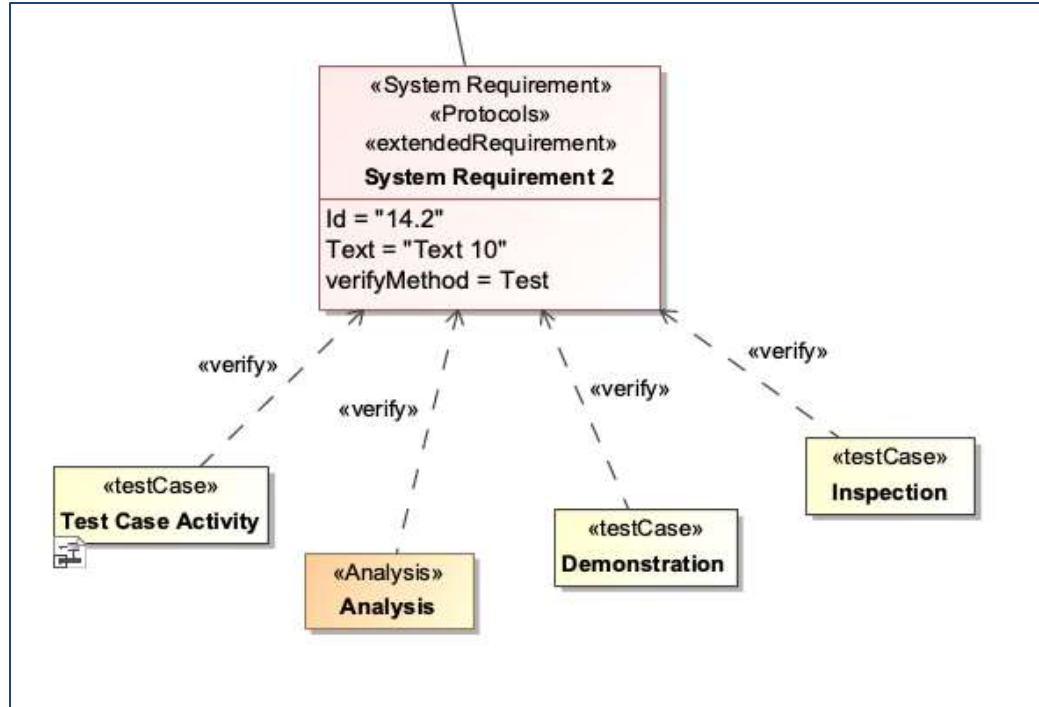
- Use case
- Other requirement
- Activity



# Relationship: *Verify*

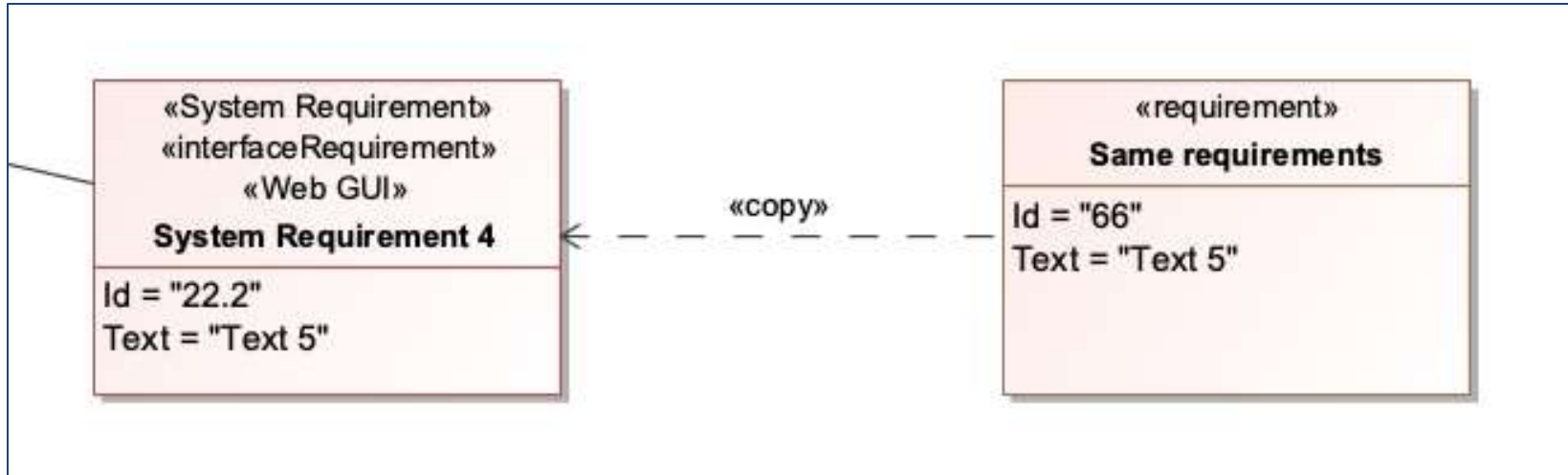
Requirement was be verified by:

- Test case/process
- Demonstration
- Analysis
- Inspection



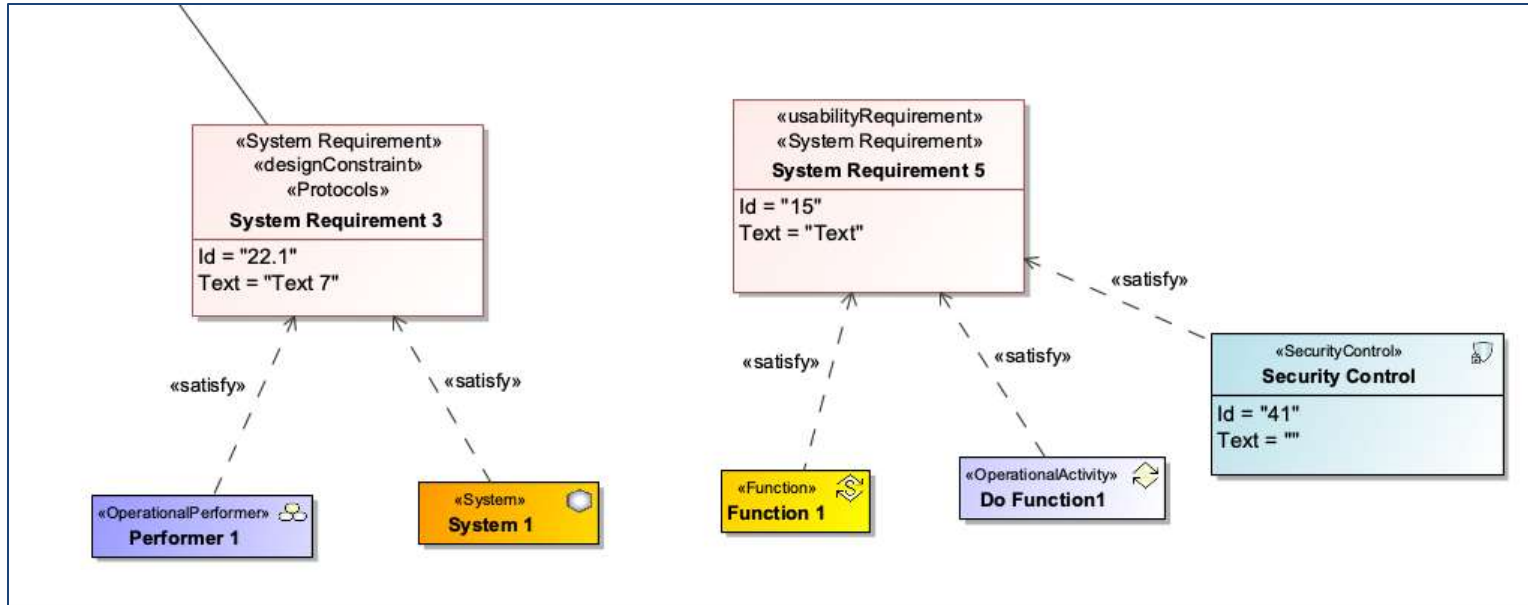
# Relationship: *Copy*

This requirement is the same as another requirement



# Relationship: *Satisfy*

Parts of the conceptual and/or solution architecture is satisfying the requirement



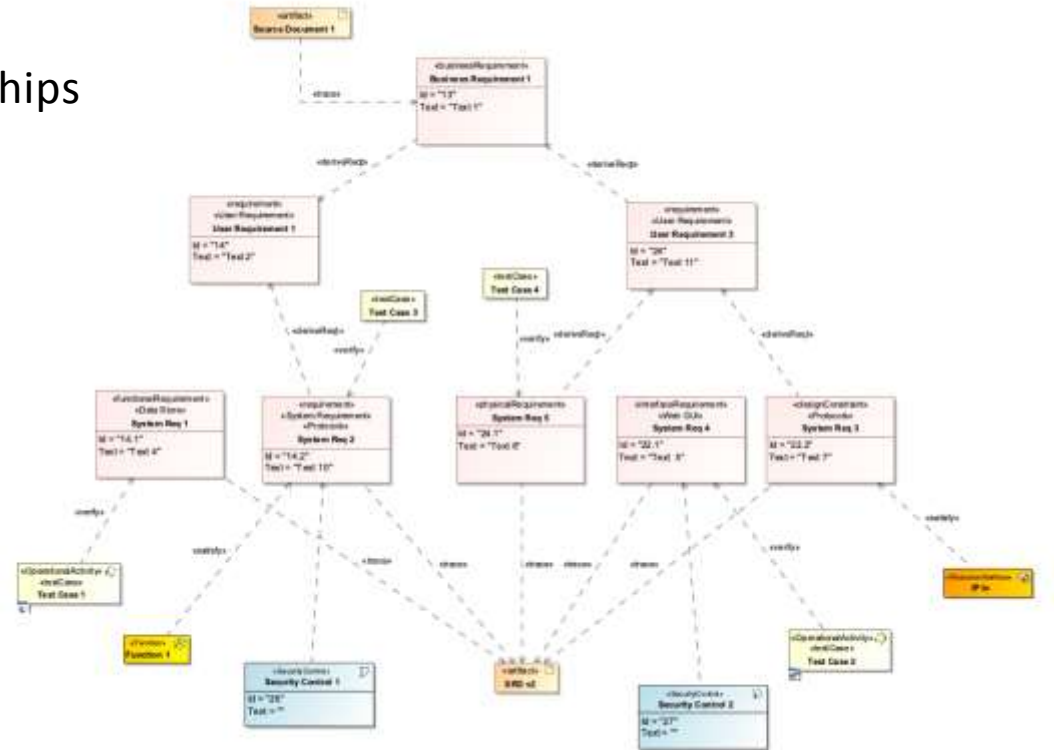
Digital Modeling Training

# Section 4: Diagrams and Analytical tools



# Requirements Diagram

- Depict all elements and relationships directly related to a requirement
- Assist with the analysis



# Requirements Table

- Traditional and convenient way to look at the requirements
- Convenient and effective way to import the requirements by bulk

Id	Name	Text	Traced To	Satisfied By	Derived From	Verified By
	Business Capabilities					
32	32 Traveler Information Management	Text 3				
32.1	32.1 Traveler Profile Management	Text 4		Traveler Profile Management System	32 Traveler Information Management	
32.2	32.2 Traveler History Management	Text 5			32 Traveler Information Management	Test Case 1
34	34 Communication Management	Text 6				
34.1	34.1 Mobile Communication Management	Text 7			34 Communication Management	
34.1.1	34.1.1 Mobile Communication Security	Text 8			34.1 Mobile Communication Management	
34.2	34.2 Optic Communication Management	Text 9			34 Communication Management	
34.2.1	34.2.1 Optic Communication Security	Text 12			34.2 Optic Communication Management	Test Case 2
35	35 Information Management	Text 13				
35.1	35.1 Information Lifecycle Management	Text 14			35 Information Management	
35.2	35.2 Information Organization	Text 15			35 Information Management	
36	36 Route Management	Text 16				
36.1	36.1 Route Analysis	Text 17			36 Route Management	
36.1.1	36.1.1 Statistical Analysis	Text 18	34 User Requirement 1		36.1 Route Analysis	
36.1.2	36.1.2 Route Visualization	Text 19	35 User Requirement 2		36.1 Route Analysis	
36.2	36.2 Route Planning	Text 20			36 Route Management	
37	37 Traffic Controls Information Management	Text 21				
38	38 Travel Condition Information Management	Text 22				
39	39 Traffic Controls Strategies Management and Optimization	Text 23				

# Dependency Matrix











- To create and show the relationships:
  - Verify
  - Satisfy
  - Derive
  - Refine
  - Trace
  - Copy

Legend		Requirements																	
↗ Satisfy																			
		Requirements <ul style="list-style-type: none"> <li>Business Requirement: 13 Business Req</li> <li>User Requirement: 14 User Req</li> <li>System: 14.1 System, 14.2 System</li> <li>User Requirement: 22 User Req</li> <li>System: 22.1 System, 22.2 System</li> <li>User Requirement: 24 User Req</li> <li>System: 24.1 System, 24.2 System</li> <li>Req Category 1: 8 Req 1, 10 Req 3, 11 Req 4, 12 Req 5</li> <li>Req Category 2: 15</li> </ul>																	
Systems Viewpoint																			
03 Element Library [SV-1]																			
IP In	1	1																	
IP Inout																			
IP Out																			
Function Library [SV-4]																			
Function 1	1	1																	
Function 2																			

Legend		Requirements																	
↗ Verify																			
		Requirements <ul style="list-style-type: none"> <li>Business Requirement: 13 Business Req</li> <li>User Requirement: 14 User Req</li> <li>System: 14.1 System, 14.2 System</li> <li>User Requirement: 22 User Req</li> <li>System: 22.1 System, 22.2 System</li> <li>User Requirement: 24 User Req</li> <li>System: 24.1 System, 24.2 System</li> <li>Req Category 1: 8 Req 1, 10 Req 3, 11 Req 4, 12 Req 5</li> <li>Req Category 2: 15</li> </ul>																	
Test Cases [Requirements]																			
Test Case 1	1	1								1									
Test Case 2	1	1																	

# Coverage Metric

- To evaluate a current state of the model
- Calculates statistics of how many requirements are covered with relationship
- Allows to monitor changes of a specific aspect of the model in time

#	 Date	 Scope	 Requirements	 Covered By Design Percentage	 Covered By Design	 Covered By Test Cases Percentage	 Covered By Test Cases
1	2020.12.04 18.49	 Requirements	15	26.6667	4	13.3333	2
2	2020.12.04 18.50	 Requirements	15	33.3333	5	13.3333	2
3	2020.12.04 18.53	 Requirements	15	26.6667	4	20	3

# Dependency Map

- Put a specific requirement at the top of the tree
- Shows all or selected types of relationships for the requirement
- Shows multiple levels of the relationships
- Shows how the requirement relates to other domain














# Find Errors:

## *Requirements Table*

- Requirements can be derived only from higher level of requirements
- In the table, there is two errors

Can you find the errors?

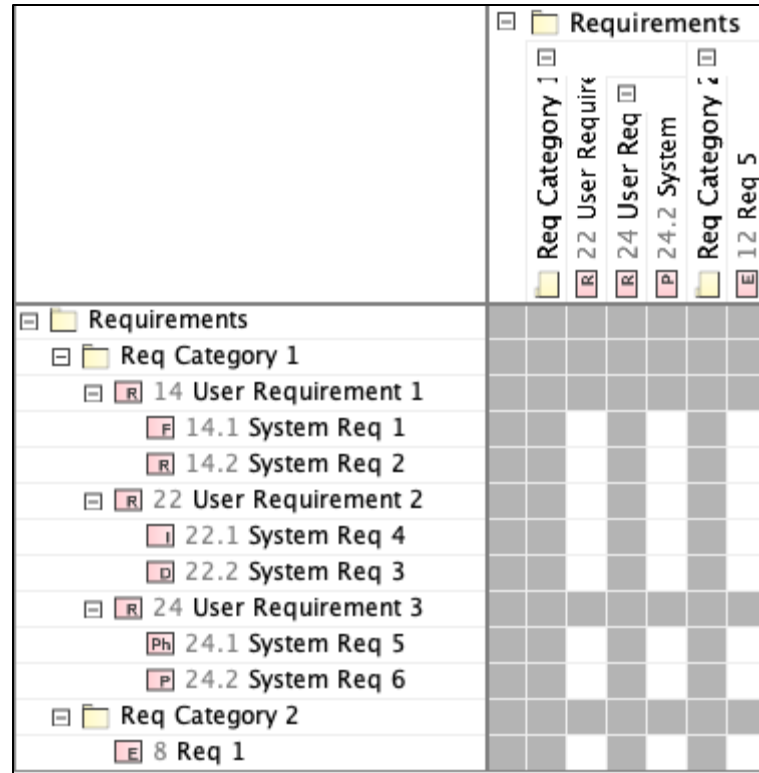
#	△ Name	Text	Derived	Derived From
1	 13 Business Requirement 1	Text 1	 24 User Requirement 3	 14 User Requirement 1
2	 14.1 System Req 1	Text 4	 14 User Requirement 1	
3	 14.2 System Req 2	Text 10		 14 User Requirement 1
4	 22.1 System Req 4	Text 5		 22 User Requirement 2
5	 22.2 System Req 3	Text 7		 24 User Requirement 3

Business requirement cannot be derived from User requirement

User requirement cannot be derived from System requirement

# Find Gaps: “Negative” Dependency Matrix

- To file a “negative” space – elements without the relationships



Digital Modeling Training

## Section 5: Demo in Cameo



