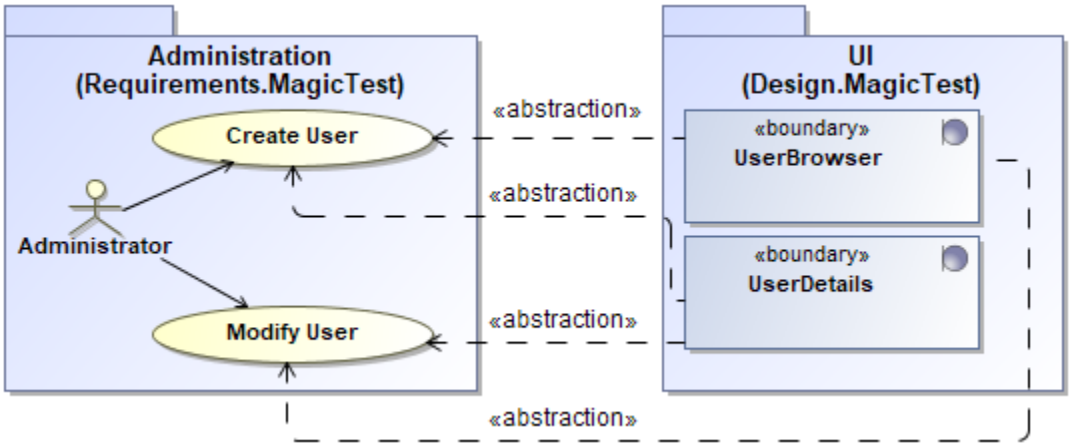
Analyzing traceability relations

You can perform the impact and/or gap analysis in your project using the Dependency Matrix feature, which is a powerful way for representing traceability relations between multiple elements from different packages, levels of abstraction, views, or other relations that cannot be represented on diagrams, for example, relations through UML tags. You can create your own dependency matrices or even custom dependency matrix types for visualization of various traceability relations. You only need to define a relevant traceability property as **dependency criteria** for this.

The following figure depicts an example of traceability relations between several model elements.



Example of traceability relations

These relations can be represented by the following traceability properties:

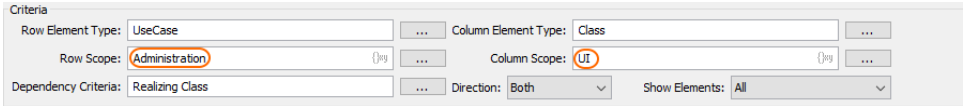
- Realizing relations (use cases classes):
  - Realizing Class
  - Realizing Element
  - All Realizing Elements
- Specifying relations (classes use cases):
  - Specifying Use Case
  - Specifying Element
  - All Specifying Elements

You can create a dependency matrix to visualize these traceability relations. The following steps will show you how to create a dependency matrix for the Realizing Class predefined property.

To create a dependency matrix for the Realizing Class predefined property:

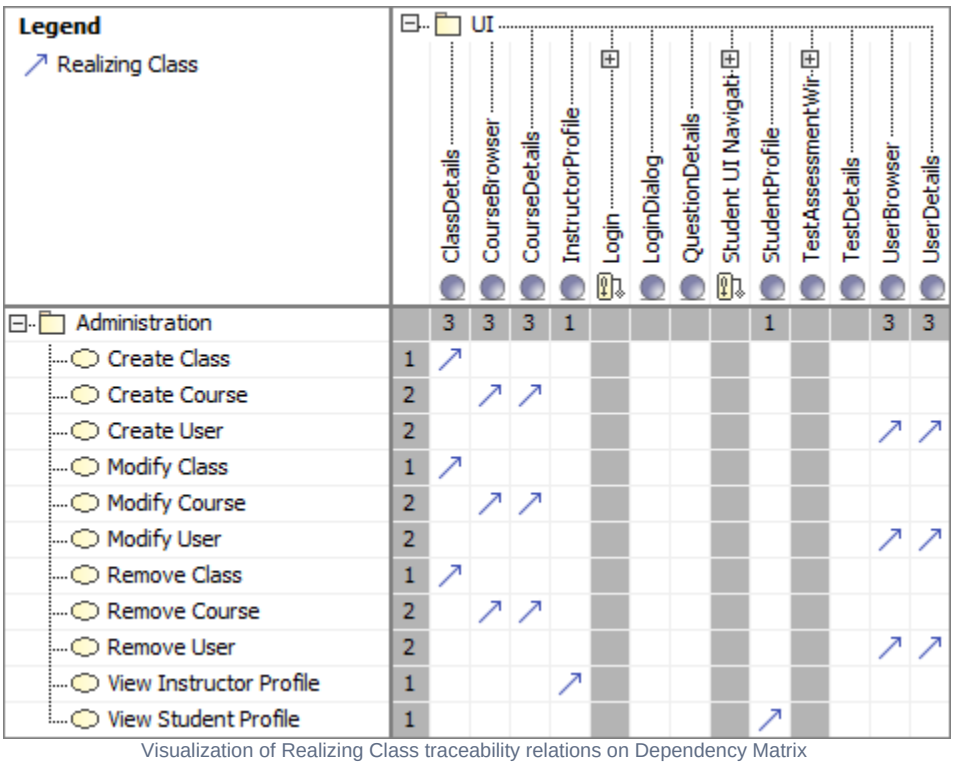
---

1. Create a Dependency Matrix diagram.
2. Define row and column type element types as follows:
   a. For the Row Element Type, select **UseCase**
   b. For the Column Element Type, select **Class**
3. Define row and column scopes as follows:
   a. For the Row Scope, click ... **> Requirements > MagicTest** and select **Administration**.
   b. For the column scope, click ... **> Design > MagicTest** and select **UI**.
4. Define the **Realizing Class** property as Dependency Criteria.
5. Rebuild the matrix (click in the diagram toolbar).



Defining criteria for visualization of traceability relationships that can be represented by Realizing Class predefined property

The following figure depicts the created Dependency Matrix showing the traceability relationships between use cases and design classes they realize:



Visualization of Realizing Class traceability relations on Dependency Matrix

Related Pages

- Traceability
  - Creating traceability relations
  - Traceability relations representation
  - Navigating between different levels of abstraction
  - Analyzing traceability relations
  - Predefined traceability rules
  - Custom traceability rules