

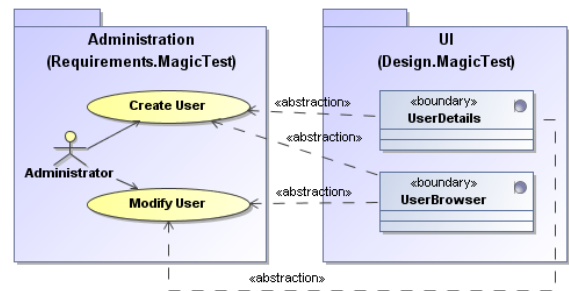
Analyzing traceability relations

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.

Related Pages

Unknown macro: 'list-c

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 type, select **"Use Case"**,
 - b. For the column type, select **"Class"**
3. Define row and column scopes as follows:
 - a. For the row scope, expand **Data > Requirements > MagicTest** and select **Administration**.
 - b. For the column scope, expand **Data > Design > MagicTest** and select **UI**.
4. Define the **Realizing Class** property as dependency criteria.
5. Rebuild the matrix.

Criteria	
Row Element Type: UseCase	Column Element Type: Class
Row Scope: Administration	Column Scope: UI
Dependency Criteria: Realizing Classes	Direction: Both

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 them realizing design classes (the highlighted intersections mark the traceability relations depicted in Figure 10).

	ClassDetails [Desi...	CourseBrowser [De...	CourseDetails [De...	InstructorProfile [De...	LoginDialog [Desi...	QuestionDetails [De...	StudentProfile [De...	TestAssessment [De...	ContributorPa...	InfoPanel [Desig...	PauseOptionPan...	PauseTimer [Desi...	TimeInfoPanel [De...	TestDetails [Desig...	UserBrowser [Des...	UserDetails [Desi...
Administration	3	3	3	1			1								3	3
Create Class	✓	✓	✓												✓	✓
Create Course		✓	✓												✓	✓
Create User															✓	✓
Modify Class	✓		✓												✓	✓
Modify Course		✓	✓												✓	✓
Modify User															✓	✓
Remove Class	✓														✓	✓
Remove Course		✓	✓												✓	✓
Remove User															✓	✓
View Instructor Profile				✓											✓	✓
View Student Profile							✓								✓	✓

Visualization of Realizing Class traceability relations on dependency matrix