

Requirements decomposition

On this page

- [Main rules of Requirements decomposition](#)
- [Decomposing Requirements](#)
 - [Creating requirements hierarchy in the Containment tree](#)
 - [Creating requirements hierarchy in the Requirement Diagram](#)
 - [Decomposing requirements in the Requirements table](#)

Main rules of Requirements decomposition

There are a few rules when you need to decompose Requirements:

- If some part of the Requirement is going to be used somewhere else, e.g. some other Requirement [derives](#) from it, or it is a subject to decomposition itself, then the requirement should be decomposed into smaller Requirements.
- If the Requirement has information from different (sub)topics, it should also be decomposed into sub-requirements.
- If parts of the Requirement are not used by any other Requirement and the Requirement holds information that naturally describes the same topic or thing, but the text of the Requirement becomes too long (e.g. 8-10 sentences), it should still be decomposed into several Requirements for better readability.

Decomposing Requirements

You can decompose Requirements in the following ways:

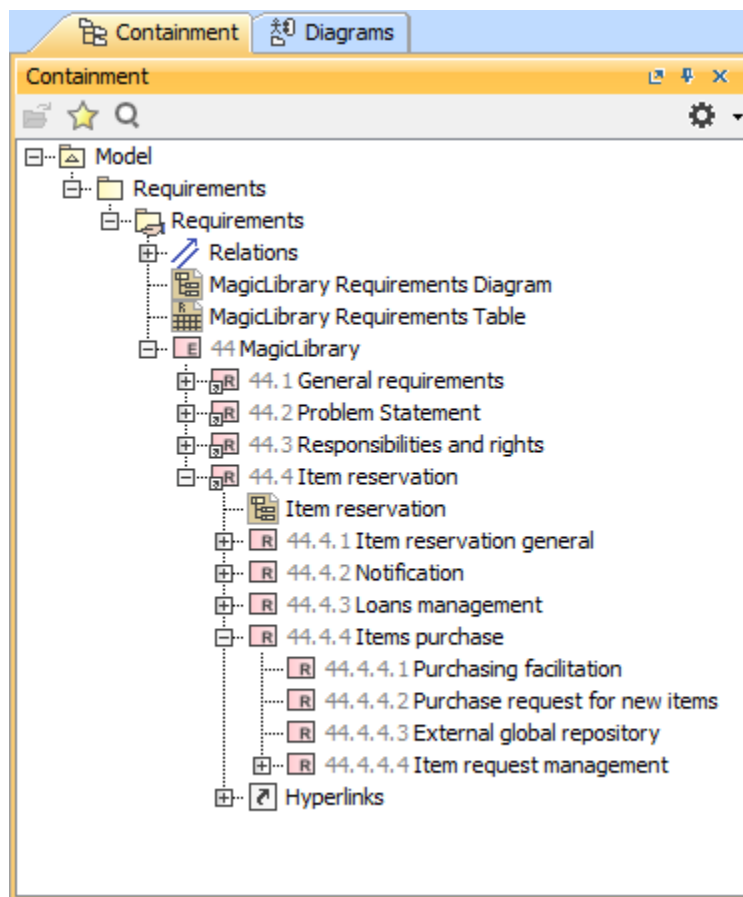
- [Creating requirements hierarchy in the Containment tree](#)
- [Creating requirements hierarchy in the Requirement Diagram](#)
- [Decomposing requirements in the Requirements table](#)

Creating requirements hierarchy in the Containment tree

You can decompose Requirements in the [Containment tree](#). The element is created in the [Model Browser](#) and then you can represent them in the [Requirement Diagram](#) by dragging elements or in [Requirement Table](#) by adding an existing elements.

[How to create requirements in the Containment tree >>](#)

In the following figure, the the [Requirements](#) are decomposed in the Containment tree. For example, the *Item purchase* Requirement is decomposed in for sub-Requirements: *Purchasing facilitation*, *Purchase request for new items*, *External global repository*, and *Item request management*.



The Requirements decomposition in the Containment tree.

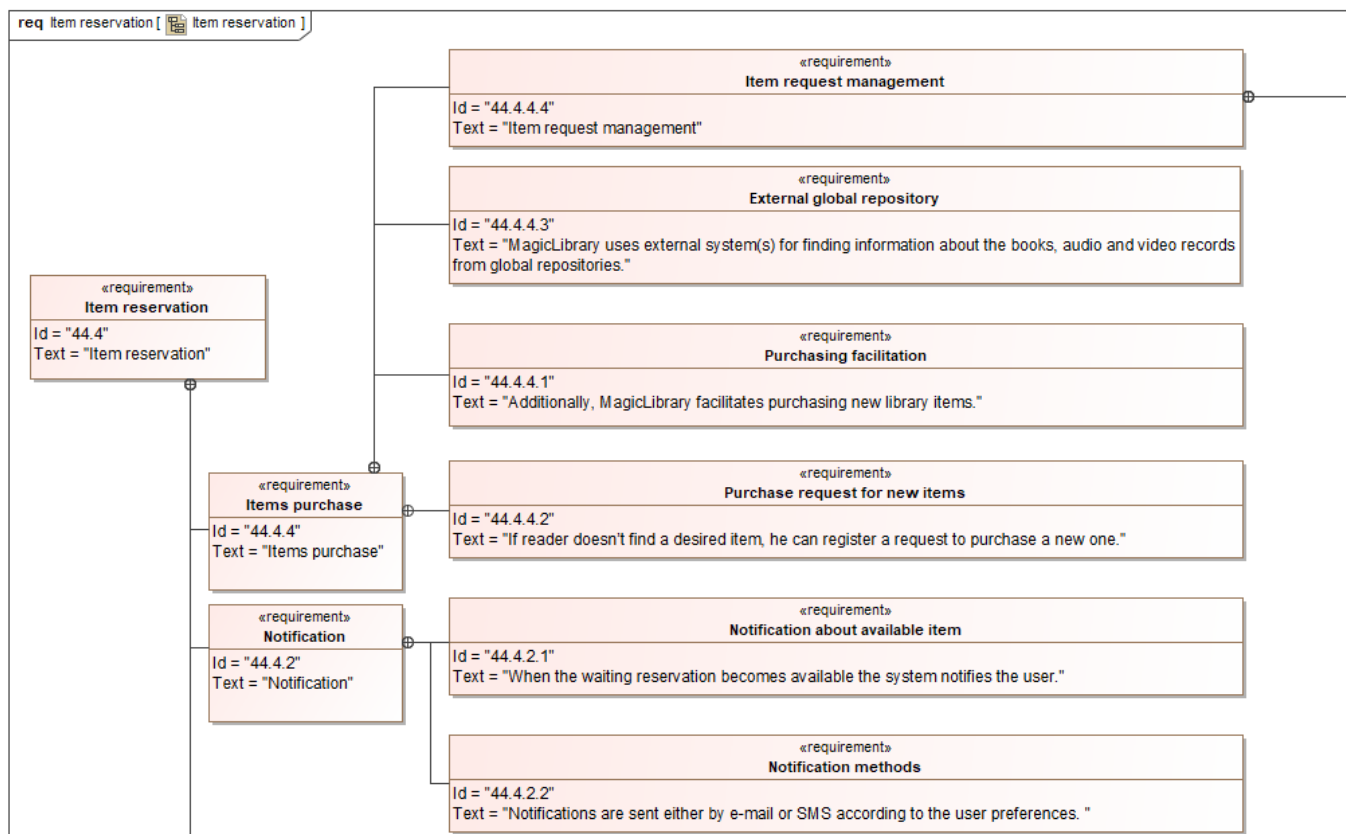
Creating requirements hierarchy in the Requirement Diagram

While decomposing [Requirements](#) in the [Requirement Diagram](#) you should follow the following rules:

- A single Requirement Diagram should contain only one nesting level: one main Requirement shape and one level of its owned Requirements.
- If each nesting level contains only a few number of Requirements (e.g. 1-4 requirements), a single Requirement Diagram can contain more than one nesting levels.
- If the main Requirement owns more than 5 Requirements they should be refined in a separate diagram.

[How to create Requirements in the Requirement diagram >>](#)

In the following figure, the the [Requirements](#) are decomposed by using the Containment relationship and represented in the [Requirement Diagram](#). For example, the *Notification* Requirement is decomposed in two sub-Requirements: *Notification about available item* and *Notification methods*.



The Requirements decomposition in the Requirement Diagram.

Decomposing requirements in the Requirements table

You can represent and create [Requirements](#) of your system in the [Requirements Table](#). [How to create Requirements in the Requirement table >>](#)

You can decompose Requirements as follows:

- [By creating Requirements in the table.](#)
- [After Requirements creation in the table.](#)

In the following figure, the [Requirements](#) are decomposed and displayed hierarchically in the [Requirement Table](#). For example, the *Item purchase* Requirement is decomposed in for sub-Requirements: *Purchasing facilitation*, *Purchase request for new items*, *External global repository*, and *Item request management*.

Requirements hierarchy mode in table

You can change the Requirements hierarchy in the table by using the **Display Mode** option. This option is described in the [Table toolbars](#) page under the Options toolbar.

#	Name	Text
1	☐ E 44 MagicLibrary	Magic Library
2	⊕ R 44.1 General requirements	General requirements
6	⊕ R 44.2 Problem Statement	Problem Statement
9	⊕ R 44.3 Responsibilities and rights	Responsibilities and rights
13	☐ R 44.4 Item reservation	Item reservation
14	⊕ R 44.4.1 Item reservation general	Item reservation general
21	⊕ R 44.4.2 Notification	Notification
24	⊕ R 44.4.3 Loans management	Loans management
29	☐ R 44.4.4 Items purchase	Items purchase
30	R 44.4.4.1 Purchasing facilitation	Additionally, MagicLibrary facilitates purchasing new library items.
31	R 44.4.4.2 Purchase request for new items	If reader doesn't find a desired item, he can register a request to purchase a new one.
32	R 44.4.4.3 External global repository	MagicLibrary uses external system(s) for finding information about the books, audio and video records from global repositories.
33	☐ R 44.4.4.4 Item request management	Item request management
34	R 44.4.4.4.1 New items requests evaluation	Librarian evaluates the registered requests for new items.
35	R 44.4.4.4.2 Requests deny	Librarian may deny the request (should provide an explanation).
36	R 44.4.4.4.3 Requests confirmation	Librarian may confirm the request, assign a priority and put it to an ordered list waiting for purchase.
37	R 44.4.4.4.4 Requests status tracking	MagicLibrary tracks the status of each request.

The Requirements decomposition in the Requirement Table.

Sample model

The model used in the figures of this page is the **User needs - requirements module for MagicLibrary** sample model that comes with MagicDraw. To open this sample do one of the following:

- Download [User needs - requirements module for MagicLibrary.mdzip](#).
- Find in modeling tool <modeling tool installation directory>\samples\diagrams\User needs - requirements module for MagicLibrary.mdzip.