

Specialization

[Go up to Terms and Definitions](#)

Discussion

A Specialization is a directed relationship between a more specific [Model](#) element and a more general Model element.

Every instance classified by the specific Model element is also classified by the general Model element. The specific Model element is therefore subject to the applicable characteristics, relationships, and constraints defined by the general Model element.

A Specialization may add characteristics or impose additional constraints. A Specialization does not remove constraints established by the general Model element.

Specialization does not mean ownership, containment, composition, association, or implementation inheritance. Those relationships have separate meanings.

The relationship is directional. A specific Model element specializes a general Model element. The general Model element does not specialize the specific Model element.

Definition

directed relationship between a specific Model element and a general Model element under which every instance classified by the specific element is also classified by the general element

Source

Adapted from:

- [OMG Systems Modeling Language \(SysML\), Version 2.0](#)
 - Part 1: Language Specification
 - Relevant concepts: Specialization, Definition, Usage, Feature, and Type
 - OMG File ID: formal/26-03-02
 - Retrieved: 2026-10-07
- 2. [OMG Unified Modeling Language \(UML\), Version 2.5.1](#)
 - Section 9.2.3.2, Generalization
 - OMG File ID: formal/17-12-05
 - Retrieved: 2026-10-07

Note

- A Specialization identifies a more specific form of a general Model element
- An instance of the specific Model element is also an instance of the general Model

element

- Constraints applicable to the general Model element also apply to the specific Model element
- The specific Model element may define additional characteristics and constraints
- A Specialization does not imply that every instance of the general Model element is an instance of the specific Model element
- A Specialization relationship may appear in a conceptual, logical, or implementation model
- A platform-independent Specialization does not require implementation using programming-language inheritance
- SysML v2 and UML may provide different concrete notations for expressing the relationship without changing its conceptual meaning

Example

[Person](#) specializes [Party](#).

Every Person is therefore a Party and is subject to the characteristics and constraints defined for Party. A Party is not necessarily a Person because Party may have other specializations, including [Organization](#).

This relationship does not require a software implementation to represent Person and Party using programming-language class inheritance.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
https://wiki.didosolutions.com/public/99_annexes/annex-b-terms-and-definitions/s/specialization

Last update: **2026/10/07 18:37**

