

Creating a necessary and sufficient condition

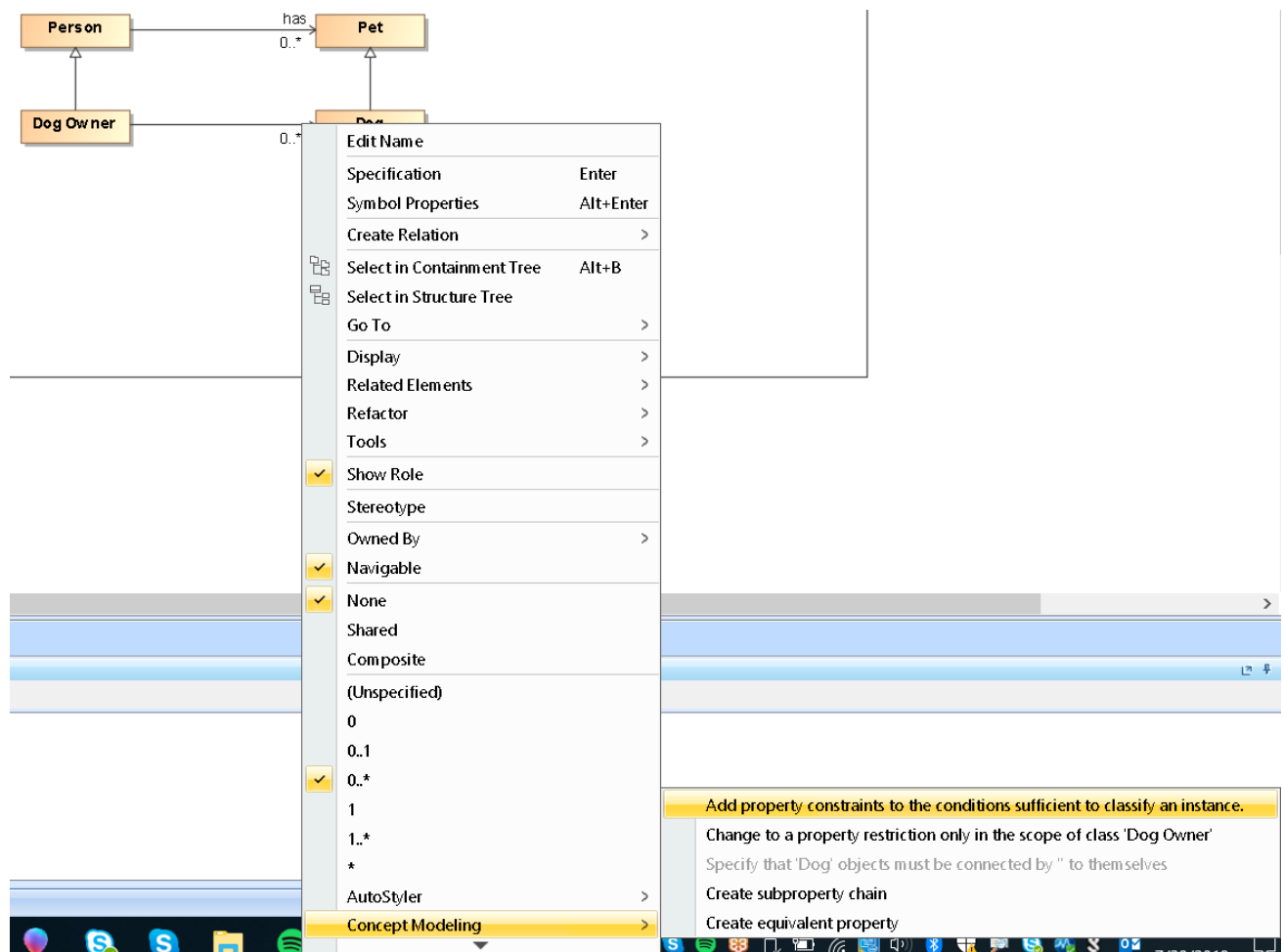
In the Concept Modeling interpretation of UML, a property that has the **Necessary and Sufficient** stereotype applied to it indicates that, when an instance satisfies the multiplicity and type constraints for the property's values, not only is it a *necessary* condition to be an instance of the class met, but a *sufficient* condition is also met (see section Condition).

Support for an Aristotelian subclass definition

Making a generalization (sufficient) provides for correct support for an Aristotelian subclass definition, where the subclass is equivalent to the intersection of all of its sufficient conditions.

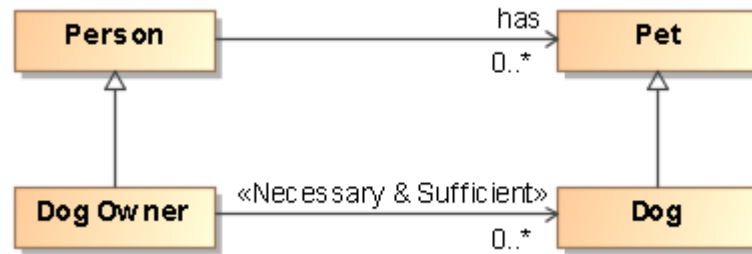
To create a sufficient condition

1. Right-click on the association end of the property to which the sufficient constraint will be applied (unnamed from “Dog Owner”). Remember that the property is owned by the class at the opposite end of the association.
2. Select **Concept Modeling > Add property constraints to the conditions sufficient to classify an instance** in the shortcut menu.



Toggling the Add property constraints to the conditions sufficient to classify an instance shortcut menu.

The **Necessary and Sufficient** stereotype is toggled on for the property.



The Necessary and Sufficient stereotype is created.

To remove a sufficient condition on a property

1. Right-click on the association end of the property of which the {sufficient} constraint will be removed (unnamed from "Dog Owner").
2. Select **Concept Modeling > Remove property constraints to the conditions sufficient to classify an instance.**

The screenshot shows the 'Tools' menu on the left, with 'Concept Modeling' expanded. The 'Remove property constraints from the conditions sufficient to classify an instance' option is highlighted. A context menu is open over the 'Dog' class, showing various actions like 'Edit Name', 'Specification', 'Symbol Properties', 'Create Relation', 'Select in Containment Tree', 'Select in Structure Tree', 'Go To', 'Display', 'Related Elements', 'Refactor', 'Tools', 'Show Role', 'Stereotype', 'Owned By', 'Navigable', 'None', 'Shared', 'Composite', '(Unspecified)', '0', '0..1', '0..*', '1', '1..*', '*', 'AutoStyler', and 'Concept Modeling'.

Toggling the Remove property constraints to the conditions sufficient to classify an instance shortcut menu.

Related page

- [Usage](#)