

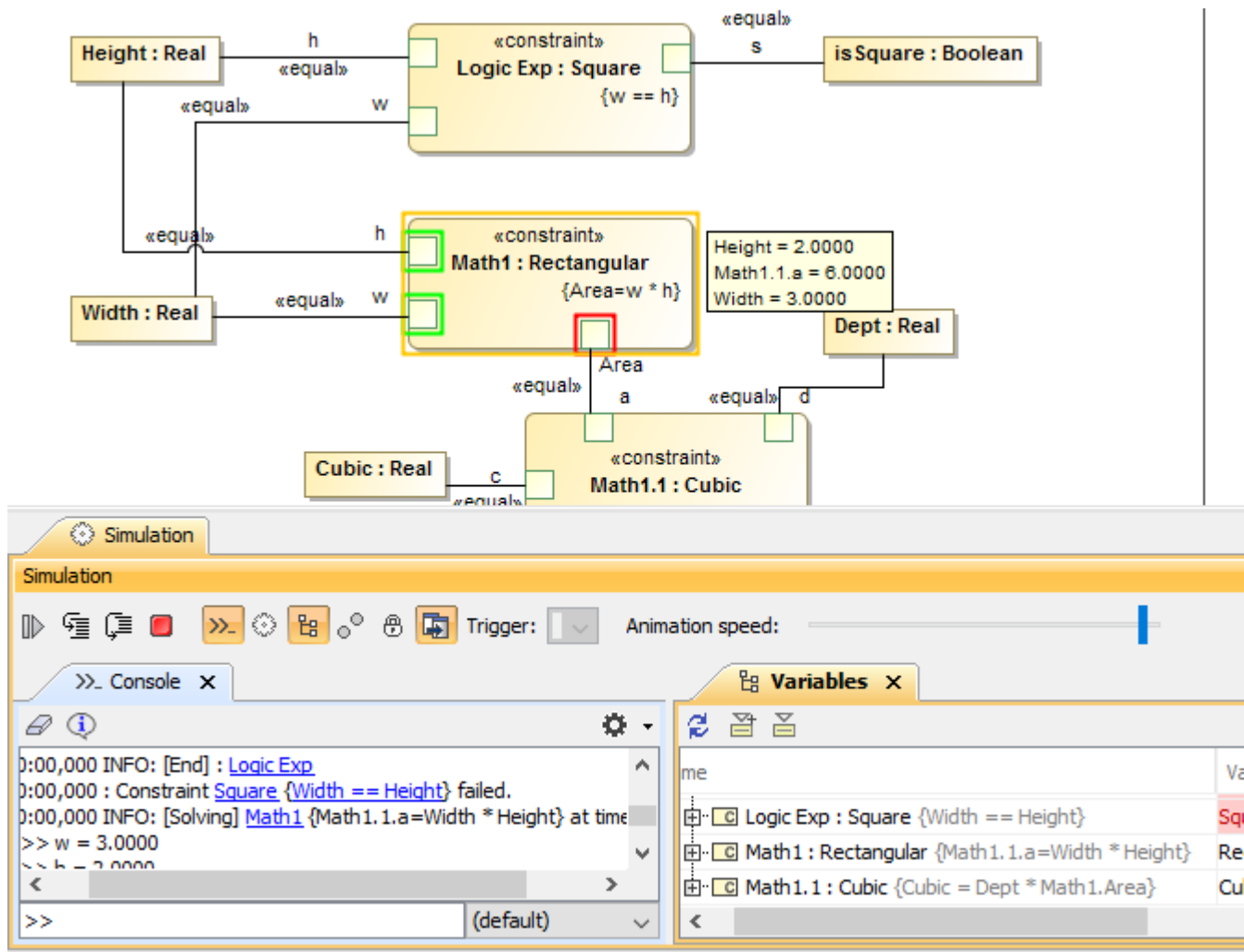
## Evaluating expressions

## On this page

- Parametric updates when States change
- Parametric updates when Activities start or end

This section explains how to evaluate the mathematical equations and logical expressions in Magic Model Analyst.

Magic Model Analyst provides constraint parameters substitution in the expressions to improve the human readability of the constraint parameters. All user interface objects (including tooltips, Console panel, and Variables panel related to a constraint property connected to a value property) will display the value property name (actual binding) instead of the constraint parameter name. This feature is applicable at the user interface level only, not at the original constraint property. For example, "Width" and "Height" (value properties) are used instead of "w" and "h" (parameter properties) in constraints, as shown in the following figure.



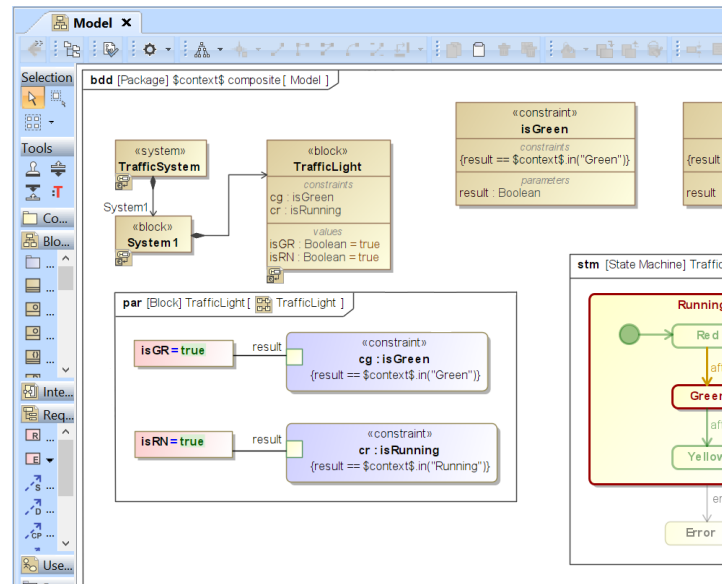
### Constraint parameters substitution in expressions.

## Parametric updates when States change

Simulation can check the object State during the evaluation. For example, if an elevator is moving, the door must be closed, or if the car is moving, the speed must be more than 0.

- You can use `$context$` (or `_context_`) in the Specification of a constraint in order to access the runtime object of the owner of the Constraint property.
- If the State check, `ALH.inState`, is used in the Specification of a constraint, the constraint will be re-evaluated on a State change with the parametric context and any objects passed as parameters into the constraint. The State change supported on simple, composite, orthogonal, and submachine.
- The Parametric update occurs after the State activation, so it depends on the selected `State Activation Semantics` option in the **Project Option** dialog:

- If **Before entry** (by default) is set, the following sequence applies: the State will be activated, the Parametric will be updated, and the Entry behavior will be simulated.
- If **After entry** is set, the following sequence applies: the Entry behavior will be simulated, the state will be activated, and the Parametric will be updated.



The constraint re-evaluated on a State change in the Parametric of the context object

## Parametric updates when Activities start or end

Simulation can check whether there is a specified Activity being executed at that time in Object\_ by using the **performs(activity String)** function, e.g., *self.performs("Activity\_1")* will return **true** when *Activity\_1* is being executed under itself and return **false** if it is not executed under other Object\_ or anywhere else. When the condition expression is evaluated at the first time, a listener for Activity execution will register the specified Activity in the *performs* ("Activity\_1") function. The listener will re-evaluate the condition expression when the context object starts or ends executing Activities.