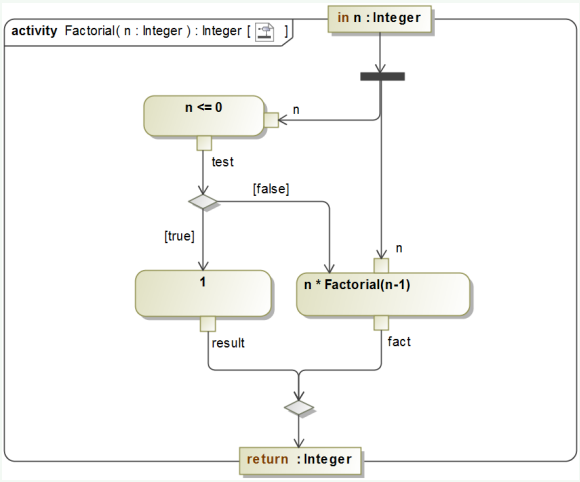
Using Alf expressions in Opaque Actions

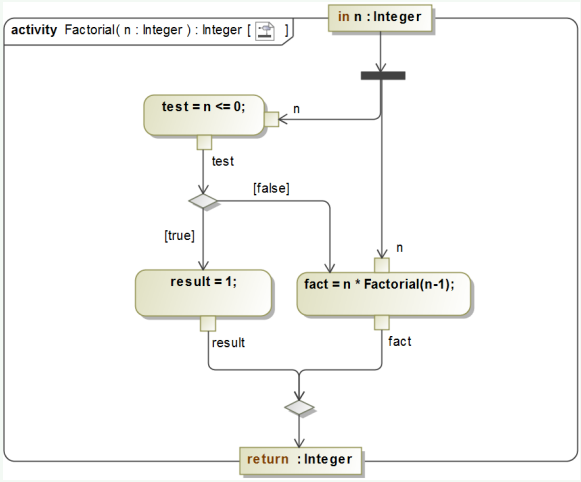
The Alf specification states that a sequence of Alf statements (each ending in a semicolon) may be used in the body of an Opaque Action. However, it is not uncommon that such an Action will simply compute a single value, to be used in later computations or as the basis for a decision. In such a case, the Alf plugin allows you to use a shortcut, in which you can use an Alf expression in the body rather than a statement. The Opaque Action must have a single Output Pin, and the type of the expression must be compatible with the type of the Output Pin. When the Opaque Action is executed, the Alf expression is evaluated and the result is placed on the Output Pin of the action.

In order to be recognized as an expression, not a statement, the Alf code entered into the Opaque Action must *not* have a semicolon at the end. If you put a semicolon at the end, then this is still valid Alf code, but it becomes an expression *statement,* and result of the expression will then *not* be automatically put on Output Pin of the Opaque Action (instead you would have to assign the value explicitly to the name of the Output Pin).

The expression shortcut for Opaque Actions is particularly useful in the context of Activities that carry out some functional computation. For example, the following Activity computes the factorial function, which is mathematically defined as *Factorial(0) = 1* and *Factorial(n) = n * Factorial(n-1)* for *n > 0.*



The above model is equivalent to the more verbose use of Alf shown in the diagram below, which does does not use the expression shortcut.



The Alf standard does not currently require support for the use of Alf expressions in Opaque Actions, so, if you create a model using the expression shortcut in MagicDraw, it may not be supported if you import your model into a tool with a different Alf implementation.

Related pages

- Using Alf in Opaque Action bodies