NULL Values

Object node values can evaluate to NULL. This value symbolizes a non-valid value of an object. For example, if a boolean value could not be calculated since an error occurred during its calculation, it might be set to NULL indicating that it is not known whether the correct value is true or false.

Undefined Attribute Values

Another application of NULL values is when class attributes and associations are undefined. If an object is created of a class that contains attributes without initial values, these attributes are set to NULL. You can use the boolean operation exists() to test whether attributes or associations exist or not.

```
aParameter.exists()
```

As an alternative, you can also use logical operators to check for NULL values, for instance within a condition() statement:

```
set s = condition (b1 != NULL, "abc", "xyz");
```

> (i) The condition() statement does not work with **DateTime** values. Use exists() in this case.

Undefined Return Values

EAL expressions may also return NULL. You can verify if operations, which explicitly can return NULL values, are doing so also by using logical operators. If you implement decisions, you could add the following conditional statement in the guard section of a transition specification:

```
aParameterString.findString("LanguageFlag") != NULL;
```