

# EAL

As former versions of UML were not sufficient to support execution of models for the purposes of testing, simulation, or full code generation, the OMG™ initiated a process that resulted in an extension to this language to include a computationally complete action model with fully defined semantics. This extension is called Action Semantics for the UML.

Action Semantics let you express actions as UML objects. An action may take a set of inputs and transform it into a set of outputs, or may change the state of the system, or both. Actions may be chained, with one action's outputs being another action's inputs. Actions are assumed to occur independently - that is, there is infinite concurrency in the system, unless you chain them or specify this in another way. This concurrency model is a natural fit to the distributed execution environment of modern enterprise and Internet applications.

To model a Service in UML, actions are defined in activity diagrams. The actions are used to configure Bridge add-ons, or to implement atomic actions like manipulating strings, arrays, and so forth. For manipulating such entities, the Bridge provides the xUML Action Language (EAL), which implements parts of the Action Semantics for the UML.

This language is used in a script like fashion in the action script part of the actions. The **Scheer PAS BUILDER** assumes that each action having the stereotype `<<ActionScript>>` contains an action script in the tagged value **script**. Expressions of the Action Language are not only used in actions but also in decision points to evaluate Boolean expressions.

Many features defined in the action semantics specification are also found in the semantics of activity diagrams, for example: iterations, decisions, signals, etc. To avoid a redundant implementation of these features, the Action Script syntax of the Bridge just supports language properties that are not - or not sufficiently - supported in activity diagrams, such as:

- Create statements
- Set and append statements (optionally with a guard)
- Object navigation
- Arithmetic and Boolean expressions
- Operations calls
- Second order operations such as: apply, reduce and select.

The following sections describe the elements of the Action Language starting with the most basic operations for variable assignment and object creation followed by a list of all built-in operations.