UBPML Syntax Overview

UBPML
Unified Business Process Modelling Language

Defines certain problem states dependent on certain conditions.

🔴	[ok]	:= { condition }
🟢	[info]	:= { condition }
🟡	[notice]	:= { condition }
🟢	[caution]	:= { condition }
	[warning]	:= { condition }
	[severe warning]	:= { condition }
	[error]	:= { condition }
	[critical error]	:= { condition }
	[fatal error]	:= { condition }
	[emergency]	:= { condition }

Defines a certain type of responsibility of this actor

«Procedure»
«Execution»
«Completion»
«Supervision»
«Overall»

A system state which is defined by a certain set of classes in certain states as well as a set of additional constraints on that classes and their relations.
A constellation is a class with stereotype "Constellation".

Constellation

Class [InState]

Class [InState]

A class in a certain state

Standard constraint { x = y ... }

Step
Attribute — Standard attribute

A step denotes a purposeful change (transformation) of the system.
A step is a class with stereotype "Step".

{ x = y ... }

Originating constellation

1

The multiplicity 1 implies that for each originating constellation instance a step instance exists.

Target constellation

Constellation

Transformation, describes a causal relation as well as an information flow - in contrast to control flow.

Steps can have multiple mutual exclusive target constellations.

Constellation

Step

Class [InState]

Classes in a certain state can be used directly to implicitly describe a certain target constellation.

{A}

In case of ambiguities, the names of the target constellations are listed as relation conditions.

{A, B}

Class [InState]

{B}

Class [InState]

Inheritance relation between steps

Inheritance relation between classes in a certain state - defines a substate.

Class [InState]

Class [InState] := { condition }

A class in a certain state along with this definition of that state. The condition is true if and only if the class is in that state.

{ActivityState = ready}

Defines initial values for the step instance

🕐 **Recurrence**
In this case, the originating constellations are defined as (recurring) points in time.

A date condition is just a problem state condition depending on timing data.

🔴🟡🟢	🕐	Date Condition
	[.....]	:= { condition }

Constellation

Class [InState]

Alternative representation of a constellation and it's defining classes.

Inheritance relation between constellations

Step

Procedure

Attaches a certain procedure to a step which is capable of performing the specified transformation. Procedures comprise manual instructions, breakdown into a sequence of substeps, or execution of automated procedures. Breakdowns comprise UBPML as well as e.g. standard BPMN or UML activity diagrams.

An instance in a certain state

Object [InState 75%]

An instance of a step. Instance diagrams are used to describe projects.

Step

Object [State 40%]

A relocation of an instance from "here" to "there" is merely a special case of a state change, but nicely visualised as a movement

here
Class

there
Class

SPHENON
The Information Architects

Author: Andreas Leue, www.leue.net