Business Process Simulation Working Group (BPSWG)

Process Analysis Data (PAF)

PAF Expression Serialization Proposals

Jeremy Horgan

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comments |
| 19 Jun 12 | 0.1 | JH | Created |
| 21 Jun 12 | 0.2 | JH/TH | * Removed the use of setInstanceParameter() as it is not required to set parameters. * Rename getInstanceParameter() to getParameter(). |
| 28 Jun 12 | 0.3 | JH/TH | * Renamed getParameter function to getProperty * Renamed InstanceParameter element to PropertyParameter * Removed roles argument from getResource * Added a getRole function * Added proposal for selectAny and an example of its usage. |
| 05 Jul 12 | 0.4 | JH | * Renamed the selectAny() function introduced in 0.3 to orResource() * Completed functions for existing distributions |
| 13 Jul 12 | 0.5 | JH | * Updated function descriptions * Removed the distribution functions. * Changed the getRole function to getResourceByRoles * Changed the signature of the getResourceByRoles to allow a variable list of role names to be specified * Replaced the example that illustrated how resources are assigned based on role. |

Proposal is to add the following XPath extensions functions to allow interchange of expressions. Expressions are required for routing decisions, setting/getting property parameter values and resource assignment.

|  |  |
| --- | --- |
| XPath Extension Function | Description / Usage |
| getProperty(name) | Returns the numeric value of a property parameter.  **Arguments**  *name*: the name of the property parameter.  **Return**  Returns a numeric value of type xs:double.  **Remarks**  If the property parameter does not exist, default to zero |
| getResource(name, qty) | Selects a collection of available resource(s) required for an Activity.  **Arguments**  *name*: the name of the resource required by the Activity. In the case of BPMN this is the attribute used to uniquely identify BPMN resource element.  *qty*: the quantity of the resource required by the Activity, expressed as an integer.  **Return**  Collection of resource(s) or an empty collection if the resource requirements were not satisfied.  **Remarks**  Resources are defined in the BPMN interchange using the <resource> element. |
| getResourceByRoles([role, …], qty) | Selects a collection of available resource(s) that can satisfy the role(s) required for an Activity. Selected resource(s) will play all roles specified by the list of roles required.  **Arguments**  *[role …]:* the variable list of required role(s).  *qty:* the quantity of a resource that satisfies the specified role(s), required by the Activity, expressed as an integer.  **Return**  Collection of resource(s) or an empty collection if the resource requirements were not satisfied  **Remarks**  A role can be applied to a resource using the PAF *role* parameter from the *ResourceParameters* perspective. |
| orResource([resources, …]) | Select the first collection of available resource(s) from the list of alternative resource(s) used for an Activity.  **Arguments**  A variable list of resources returned by the getResource() or getResourceByRoles() functions.  **Return**  Collection of resource(s).  **Remarks**  This allows alternative behaviour for resource selection. The evaluation order of resources is from left to right. |

# Property Parameter used for Processing Time



Expression: “Volume \* 6”

<ElementParameters elementId="Task1">

<TimeParameters>

<ProcessingTime>

<ExpressionParameter value="paf:getProperty('Volume') \* 6"/>

</ProcessingTime>

</TimeParameters>

</ElementParameters>

# Property Parameter Assignment

Expression: “Volume = (Volume \* 5) / 100”

<ElementParameters elementId="Start">

<PropertyParameters>

<Property name="Volume">

<ExpressionParameter

value="(paf:getProperty('Volume') \* 5) div 100"/>

</Property>

</PropertyParameters>

</ElementParameters>

Expression: “Volume = Uniform(60, 600)

<ElementParameters elementId="Start">

<PropertyParameters>

<Property name="Volume">

<UniformDistribution min="160" max="600"/>

</Property>

</PropertyParameters>

</ElementParameters>

# Property Parameter Decision Routing

Expression: “Volume \* 10 > 100”

<ElementParameters elementId="Flow1">

<ControlParameters>

<Condition>

<ExpressionParameter

value="paf:getProperty('Volume') \* 10 &gt; 100"/>

</Condition>

</ControlParameters >

</ElementParameters>

# Resource Assignment

Expression: “10 \* developer” resources

<ElementParameters elementId="Task1">

<ResourceParameters>

<Selection>

<ExpressionParameter

value="paf:getResource('developer', 10)"/>

</Selection>

</ResourceParameters>

</ElementParameters>

Expression: “2 \* developer and 1 Sales” resources. Uses the XPath UNION operator.

<ElementParameters elementId="Task1">

<ResourceParameters>

<Selection>

<ExpressionParameter

value="paf:getResource('developer', 2) | paf:getResource('sales', 1)"/>

</Selection>

</ResourceParameters>

</ElementParameters>

Expression: “1 \* developers or 5 \* Sales” resources

<ElementParameters elementId="Task1">

<ResourceParameters>

<Selection>

<ExpressionParameter

value="paf:orResource(

paf:getResource('developer', 1),

paf:getResource('sales', 5))"/>

</Selection>

</ResourceParameters>

</ElementParameters>

Expression: “1 \* resource with a **lead role** and a **java role**”

<ElementParameters elementId="Task1">

<ResourceParameters>

<Selection>

<ExpressionParameter

value="paf:getResourceByRoles('java role', 'lead role', 1)"/>

</Selection>

</ResourceParameters>

</ElementParameters>