Enterprise Architect

User Guide Series

Business Process Simulation (BPSim)

Author: Sparx Systems
Date: 15/07/2016
Version: 1.0

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Business Process Simulation (BPSim)

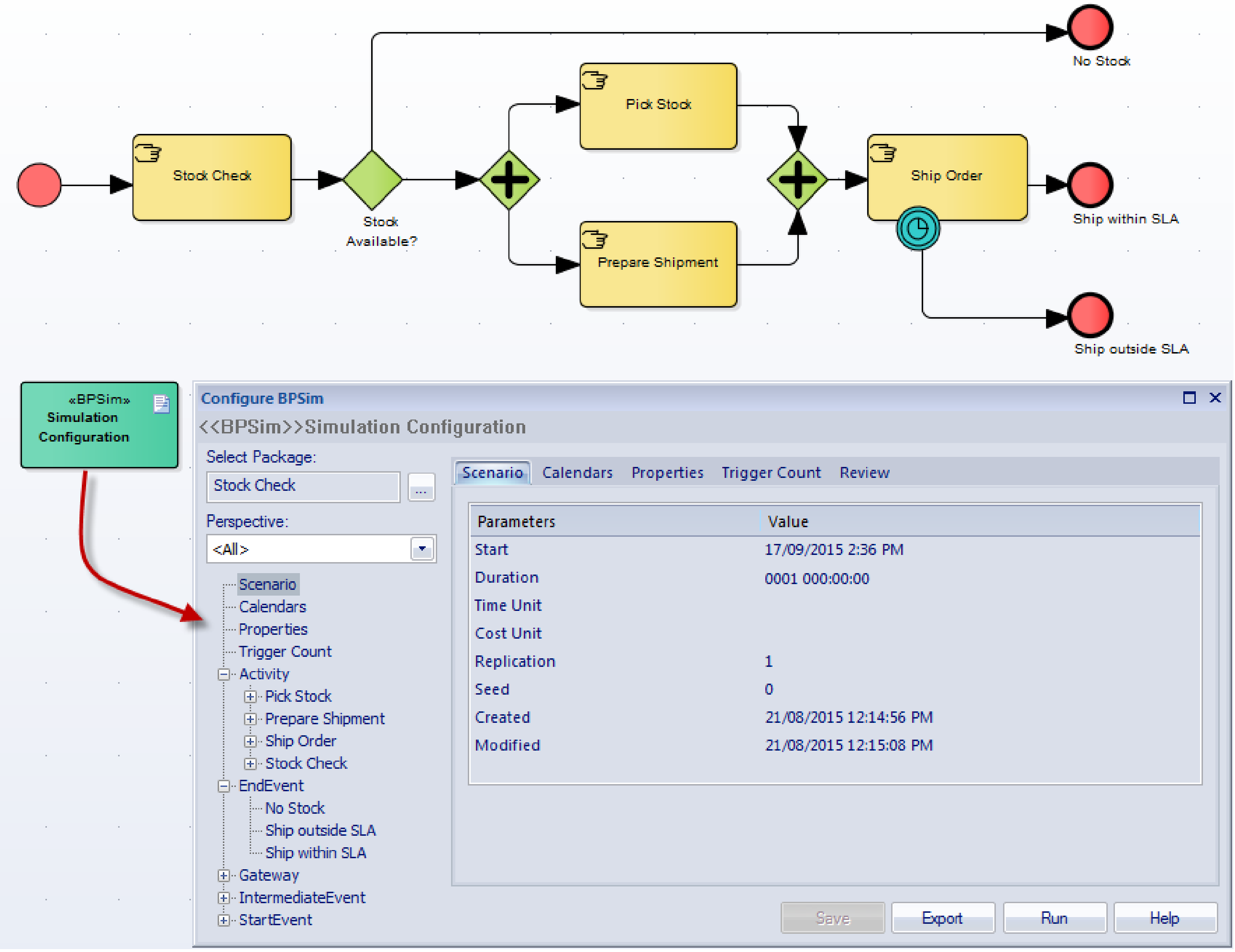
The open BPSim specification provides a rich set of material on how to configure and assign resources to activities/tasks, how to raise events, decision making and other real-world capabilities. Once configured according to the BPSim specification, a business process model (constructed in BPMN) can be sent to a suitable BPSim simulation engine and run according to the process defined in the BPMN model, using the configuration data attached in the BPSim information.

The BPSim specification is very detailed and offers the interested modeler and business strategist an unprecedented flexibility in assigning operating information to a model and then assessing the quality of the solution based on information received back from the Simulation engine. This section describes in detail the various screens and options available when configuring a model for BPSim execution.

Sparx Systems provide a BPSim-capable simulator (for separate purchase) - the **MDG BPSim Execution Engine**. This **Add-In** integrates with the BPSim and BPMN models defined in Enterprise Architect, providing the capability to run and store the results from multiple simulations and to perform convenient comparisons across each configuration's result set.

Installing the **MDG BPSim Execution Engine** and registering the licence for it are pre-requisites for accessing and using the BPSim configuration facilities. Once you have set up a BPSim configuration, you can export the BPMN model along with its BPSim data in a standard form that can be consumed by the Sparx Sysytems BPSim Execution Engine and by any other standards-compliant BPSim engines.

BPMN Model with BPSim Configuration



<<BPSim>>Simulation Configuration

Select Package: Stock Check

Perspective: <All>

Parameters	Value
Start	17/09/2015 2:36 PM
Duration	0001 000:00:00
Time Unit	
Cost Unit	
Replication	1
Seed	0
Created	21/08/2015 12:14:56 PM
Modified	21/08/2015 12:15:08 PM

The BPMSim configuration screens have been divided into multiple **Perspectives**, each of which focuses on one aspect of the setup process at a time. The three Perspectives are:

- Control Perspective, which examines how activity flows through the process, moderated by the likelihood of a sequence of events and the priorities of certain events
- Time (Temporal) Perspective, which examines how the **duration** of one or more phases in the processing of an Activity influences the business process
- Resource Perspective, which examines the involvement of types and roles of workers and other resources, their required numbers, their costs and their availability

A fourth Perspective (<All>) presents all of the options of the other three Perspectives.

You can also maintain multiple versions of a configuration and easily compare the differences between versions to see how each configuration will vary the flow of the proposed simulation or process execution. You might, for example, establish a baseline configuration and then create multiple 'what-if?' configurations that vary one or more parameters. Once run through a simulation engine you can examine each result and decide on the relative merits of each configuration. One useful facility to apply here is the simple inheritance of common, unchanged data in one configuration by another configuration that contains only the data being varied.

Users can combine the BPSim and Charting facilities to quickly vary, simulate and compare aspects of a Business Process model, and show the differences between the simulations in one of the many Chart formats.

If you are working across multiple projects, you can export and import the BPSim configurations between them. The configuration automatically carries with it the BPMN 2.0 model on which it is based.

The Enterprise Architect Business Process Simulation configuration tool is based on the BPSim Framework developed by the Workflow Management Coalition (WfMC).

Notes

- If you click on a business process element or connector in a diagram or in the **Project Browser**, it is highlighted and selected in the 'Configure BPSim' dialog
- The Business Process that you simulate can contain elements from more than one Package; to include the external elements in the simulation, you must create a Package diagram containing the 'parent' Package and either the 'external' Packages containing the external elements, or the external elements themselves; create a:
  - Package Import connector from the parent Package to each external Package, or
  - Dependency connector from the parent Package to each external element

Working in BPSim

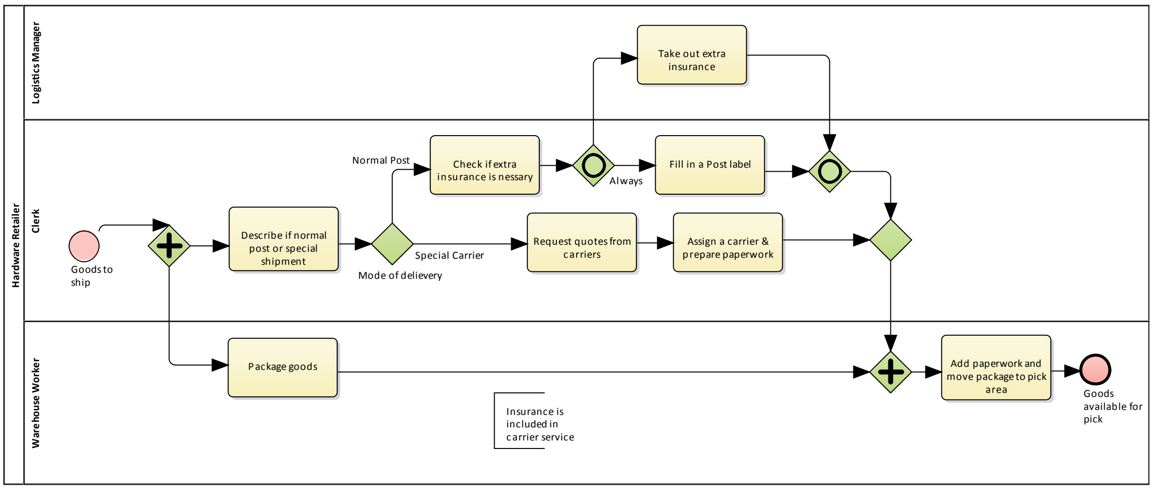
A Business Process Simulation (BPSim) configuration is represented and contained in a **Business Process Simulation Artifact** element, which you can create on a diagram in any Package in the same project as the BPMN model that you are working with.

Create a Business Process Model

Each BPSim configuration is created specifically for and from an existing Business Process, defined in BPMN. Therefore you will need to create or import the model on which the configuration is to be based, before you use the **Business Process Simulation Artifact**. As you develop the configuration, you can continue to edit the model; changes are immediately reflected in the configuration.

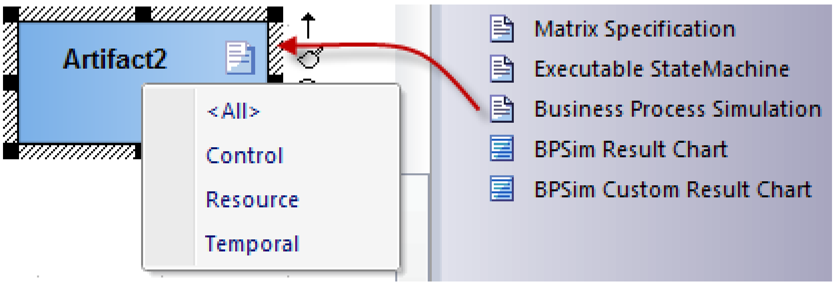
This example diagram can be found and worked on in the EAExample model, in:

Analysis and Business Modeling > BPMN 2.0 Examples > Process Diagrams > Shipment Process of a hardware retailer


Hardware Retailer
Logistics Manager
Clerk
Warehouse Worker
Take out extra insurance
Normal Post
Check if extra insurance is nessary
Always
Fill in a Post label
Goods to ship
Describe if normal post or special shipment
Special Carrier
Mode of delievery
Request quotes from carriers
Assign a carrier & prepare paperwork
Package goods
Insurance is included in carrier service
Add paperwork and move package to pick area
Goods available for pick

Create a Business Process Simulation Artifact

Open a diagram in which to create the Artifact, and display the **Diagram Toolbox** (press **Alt+5**). Expand the common 'Artifacts' page, and drag the 'Business Process Simulation' icon onto the diagram.



A prompt displays from which you can select to create the BPSim Configuration from the perspective of Control, Resource, Temporal (Time), or all three. Select the appropriate option.

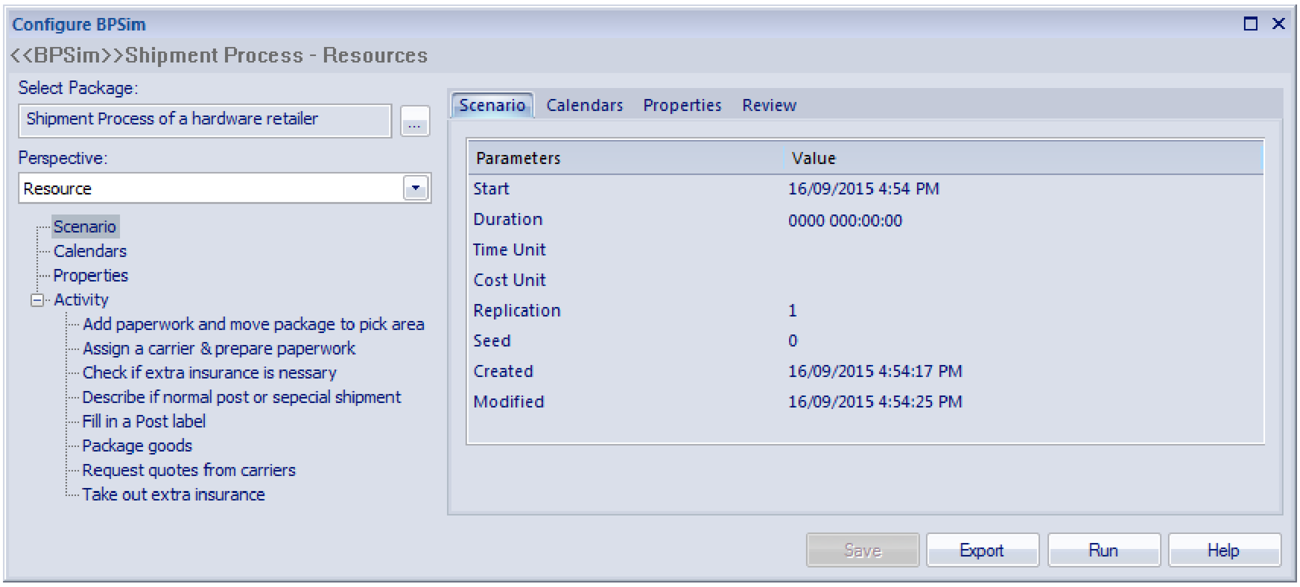
When you set up the Artifact, consider whether you might create separate Artifacts for the different **Perspectives** and use

Generalization connectors between them so that one Artifact inherits the data you have defined in another Artifact. This way, you do not have to keep re-defining the whole configuration in every Artifact you create.

Double-click on the element and give it an appropriate name.

Create the Business Process Simulation Configuration

Right-click on the Artifact element (either in the diagram or in the **Project Browser**) and select the 'Configure BPSim' option. The **Configure BPSim window** displays for the Artifact. The window initially contains no data; to initialize it, click on the **Browse button** to the right of the 'Select Package' field, and browse for the Package containing the Business Process for which to create the simulation configuration. The window is then populated with the names of the objects in the Business Process, and fields appropriate to the configuration for those objects.


Configure BPSim

<<BPSim>>Shipment Process - Resources

Select Package: Shipment Process of a hardware retailer

Perspective: Resource

- Scenario
- Calendars
- Properties
- Activity
  - Add paperwork and move package to pick area
  - Assign a carrier & prepare paperwork
  - Check if extra insurance is nessary
  - Describe if normal post or sepecial shipment
  - Fill in a Post label
  - Package goods
  - Request quotes from carriers
  - Take out extra insurance

Parameters	Value
Start	16/09/2015 4:54 PM
Duration	0000 000:00:00
Time Unit	
Cost Unit	
Replication	1
Seed	0
Created	16/09/2015 4:54:17 PM
Modified	16/09/2015 4:54:25 PM

The Duration parameter is a required value. It must be long enough to accommodate a complete simulation; for example, if a process (and hence its simulation) takes three hours to complete, the Duration parameter must be set to a value greater than three hours.

The objects and fields shown are determined by the Perspective you selected when creating the BPSim Artifact; if you want to change this Perspective, click on the drop-down arrow in the 'Perspective' field and select the new value.

Note that you can dock the window either to one edge of the screen or as a View in the central work area.

Define Control Perspective

To begin defining a simulation, select the 'Control' Perspective. This Perspective helps you to focus on the flow of control through your simulation, running from the Start event to the End event.

1. Select the Start Event in either the BPSim Configuration, a diagram or the **Project Browser**.

2. The 'Control' tab displays on the right hand side of the configuration.

3. Select the 'New Parameter...' combo and select 'TriggerCount'. Type in the number of times the simulation path will be triggered.
4. Select the 'New Parameter...' combo again and select 'InterTriggerTimer'. Type in the **duration** - the length of time paused between triggers.

Now you can configure the sequence flow of control through Gateways. Control is set on SequenceFlow relationships that flow from a Gateway as source to a target when more than one path is viable.

1. Select the SequenceFlow relationship in either the BPSim Configuration, a diagram or the Project Browser.
2. The 'Control' tab displays on the right hand side of the configuration.
3. Select the 'New Parameter...' combo and select 'Probability'. Probability is set by a decimal portion of 1.0; the Probabilities of all the relationships add up to 1.0.
4. Set the 'Probability' value for each of the SequenceFlow relationships issuing from the Gateway.

Continue through your BPMN diagram, setting the Control Perspective elements for this simulation.

Learn more

- The BPSim Control Perspective

Define Temporal Perspective

To begin to define the Temporal Perspective of the simulation, select 'Temporal' in the 'Perspective' field. This Perspective helps you to focus on setting the length of time taken to complete an Activity task.

5. Select an Activity in the BPSim Configuration, a diagram or the **Project Browser**.
6. The 'Time' tab displays on the right hand side of the configuration.
7. Select the 'New Parameter...' combo and select 'ProcessingTime'. Enter the Duration time value for this Activity.

Continue through your BPMN diagram setting the Temporal Perspective elements for this simulation.

Learn more

- The BPSim Time Perspective

Define Resource Perspective

To begin defining the Resource Perspective of the simulation, select 'Resource' in the 'Perspective' field. This Perspective helps you to focus on defining the resources required to complete an Activity.

Note: Your Activities might already have ResourceRoles defined, but for this explanation we will assume none are defined.

8. Select an Activity in the BPSim Configuration, a diagram or the **Project Browser**.
9. The 'ResourceRole' tab displays on the right hand side of the configuration.
10. Click on the **Add Resource Role button** to add a ResourceRole element to your Activity as a child element in the Project Browser.
11. Select the ResourceRole list item added to the 'Resource Roles defined in BPMN' list.
12. Click on the 'BPMN Resource' list item **Browse button** to select a project-defined 'Resource' element, or create a new resource using the **Add New button** in the 'Selection' dialog.
13. Select the 'Quantity' field and enter the number of resources required to complete this single task.

A Resource has now been configured for this Activity. The next step is to ensure that the total pool of this resource has been configured.

1. Select the defined 'Resource' element in the BPSim Configuration, a diagram or the Project Browser.
2. The 'Resource' tab displays on the right hand side of the configuration.

3. Select the 'New Parameter...' combo, select 'Quantity', and type in the total amount of this resource in the pool available to the simulation for any given task.

Continue through your BPMN diagram setting the Resource Perspective elements for this simulation.

Running BPSim

You should now have enough parameters configured to run a simple BPSimulation.

1. Click on the **Run button**. The 'BPSim Simulation Controller' dialog displays.
2. Click on the Run button and select either 'Standard Simulation' or 'Customized Simulation'.
3. The simulation executes.
4. The runtime token count is displayed in the bottom pane of the dialog.
5. Click on the **Open Result button** to view the simulation results.

Learn more

- Business Process Simulation (BPSim)
- Using the Parameter Value Dialog
- The BPSim User Interface Components
- The BPSim Resource Perspective
- Configuration Inheritance
- Using the MDG BPSim Execution Engine
- Tracking Property Parameter Values
- Compare BPSim Configurations
- Export a BPSim Configuration

Using the Parameter Value Dialog

The 'Parameter Value' dialog helps you to define values for a wide range of parameters throughout the BPSim Configuration. It supports the definition of simple fixed values through to distributions and expressions that yield a derived value. Not all types of value or derivation are appropriate to all types of parameter.

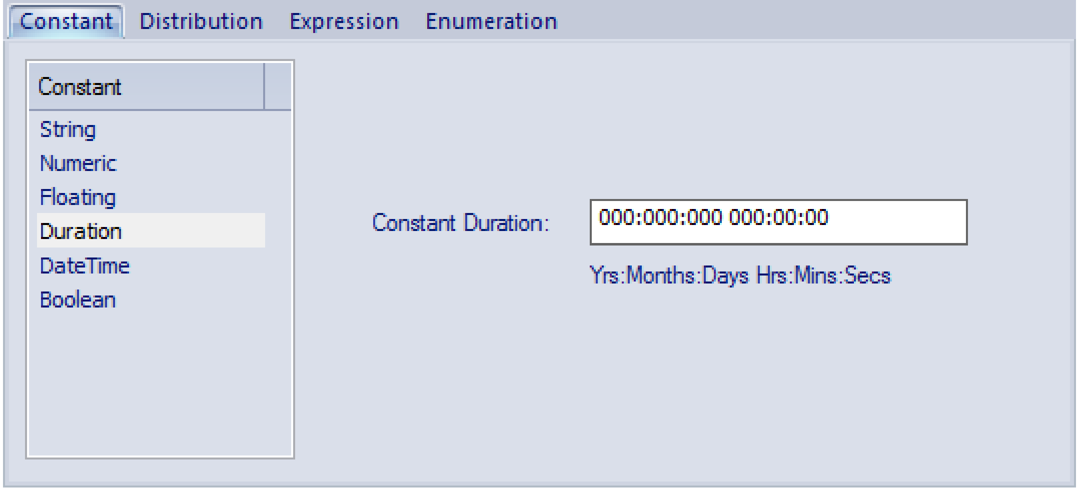
The dialog name is taken from the object type, object name and the name of the parameter being defined; for example, <<Activity>>Resolve Issue : ProcessingTime.

Access

With a BPSim artifact loaded in the 'Configure BPSim' window |select a Perspective |select an object in the lower left pane | select a tab page in the right-hand pane | select a parameter in the 'Type' field |click on ... in 'Value' field

Constant Tab

Use this tab to define a specific value for the parameter - a numeral, text string or time, for example.



Constant Duration: 000:000:000 000:00:00

Yrs:Months:Days Hrs:Mins:Secs

In the 'Constant' panel, select the type of constant:

- Floating
- Numeric
- String
- DateTime
- Boolean, or
- Duration

Appropriate fields display to the right of the panel; type the value and, if required, the unit in which the value is expressed (for example, a unit of time or of currency). For some types of parameter a drop-down list is available from which you can select a value.

Distribution Tab

On this tab, you can apply a statistical sampling method to obtain the parameter value; for each type of distribution available, the appropriate fields display for you to enter the parameters of the distribution. All distributions require you to identify the unit of expression.

The distribution parameters are not necessary for the business process you are developing, but (if you are deriving values from a distribution) are required for the simulation.

You can select from these types of distribution:

- **Beta** - a continuous probability distribution providing 'real' values within a short range, commonly 0 to 1
- **Weibull** - a continuous probability distribution providing 'real' values, commonly used for object lifetime analysis
- **Gamma** - a continuous probability distribution providing 'real' values, useful for modeling exponentially distributed random variables
- **Binomial** - an 'integer' distribution, providing values based on the number of trials and the probability of a certain outcome
- **Erlang** - provides 'real' values based on the Erlang *K* value and the mean of the distribution
- **Normal** - provides 'real' values based on the mean and standard deviation of the distribution
- **LogNormal** - a continuous probability distribution of 'real', random variables whose logarithm is normally distributed
- **Poisson** - a discrete ('integer') probability distribution that expresses the probability of a given number of events occurring independently in a fixed interval of time or space (volume, distance or area)
- **NegativeExponential** - provides 'real' values based on the mean of the distribution
- **Triangular** - provides 'real' values based on the mode of the distribution and the minimum and maximum values of a range
- **TruncatedNormal** - provides 'real' values based on the mean and standard deviation of points within the minimum and maximum values of a range
- **Uniform** - provides 'real' values between the minimum and maximum values in a range

Expression Tab

On this tab, you type in an XPATH 1.0 expression to combine explicit values, operators and functions to be processed at runtime to provide a value. Each component of an expression must be enclosed in braces - {xxx}. For example:

n = {a^2} x {b^2}

Enumeration Tab

On the 'Enumeration' tab, you can define an enumeration to supply a collection of constant values. You would have obtained these values from real-world, historical data or from analysis and simulation of a model. Every time the parameter is evaluated, the next enumeration value is returned.

As you define each numeration value, click on the **Save button** to add it to the list of possible values, and click on the **New button** to clear the data fields ready to enter another value. For some types of enumeration value you might be asked to define the unit in which the value is expressed. The types of enumeration you can define include:

- String
- Floating
- Numeric
- Duration
- DateTime
- Boolean

The BPSim User Interface Components

The BPSim **Perspectives** have four pages, or tabs, of data in common. These are the 'Scenario', 'Calendars', 'Properties' and 'Review' tabs, the first three of which are also reflected in the data hierarchy on the left of the window. The data specific to each Perspective is discussed in separate topics.

The **Save button**, **Validate** Button, **Export button** and **Run button** are also available across the configuration, to save all changes made, to validate the configuration, to export the configuration and to execute the Sparx Systems Business Process Simulation engine (if available) on the current configuration.

Because these tabs are in common, you might prefer to populate them with data in a 'base' version of the configuration and use inheritance to make that data available to other configurations. This helps reduce the work in defining the data in the first place, and avoids errors in repeatedly entering the same information in different configurations - you only have to enter it once.

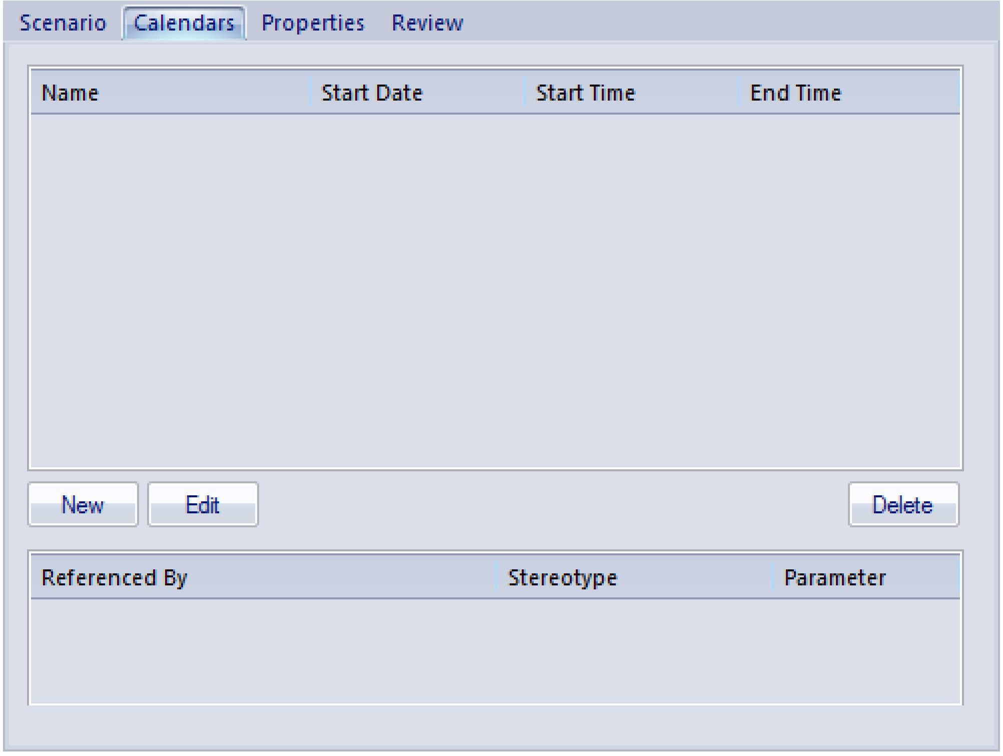
The Scenario Tab

This tab defines the general characteristics of the scenario that the configuration represents, such as:

- The date and time at which the process takes effect; you set and edit this by overtyping the values or, for the date, by selection from a drop-down calendar
- The length of time the process takes; overtype the appropriate segment with the number of years:days:hours:minutes:seconds:milliseconds
- The base unit in which periods of time are expressed in this scenario (unless overridden); click on the drop-down arrow and select the unit
- The currency unit of any costs recorded in the process; click on the drop-down arrow and select the unit abbreviation
- The number of times the scenario as a whole will be repeated in an execution cycle, which you simply type in
- The 'seed' or starting number for any sequence of randomly-generated numbers in the process, which you simply type in
- The date and time on which the scenario was created
- The date and time at which the scenario was last modified
- The JRE Home directory, which you can browse for and select
- The JDK Home directory, which you can browse for and select
- The number of the Port through which Enterprise Architect communicates with the BPSim Execution Engine; this defaults to 1799

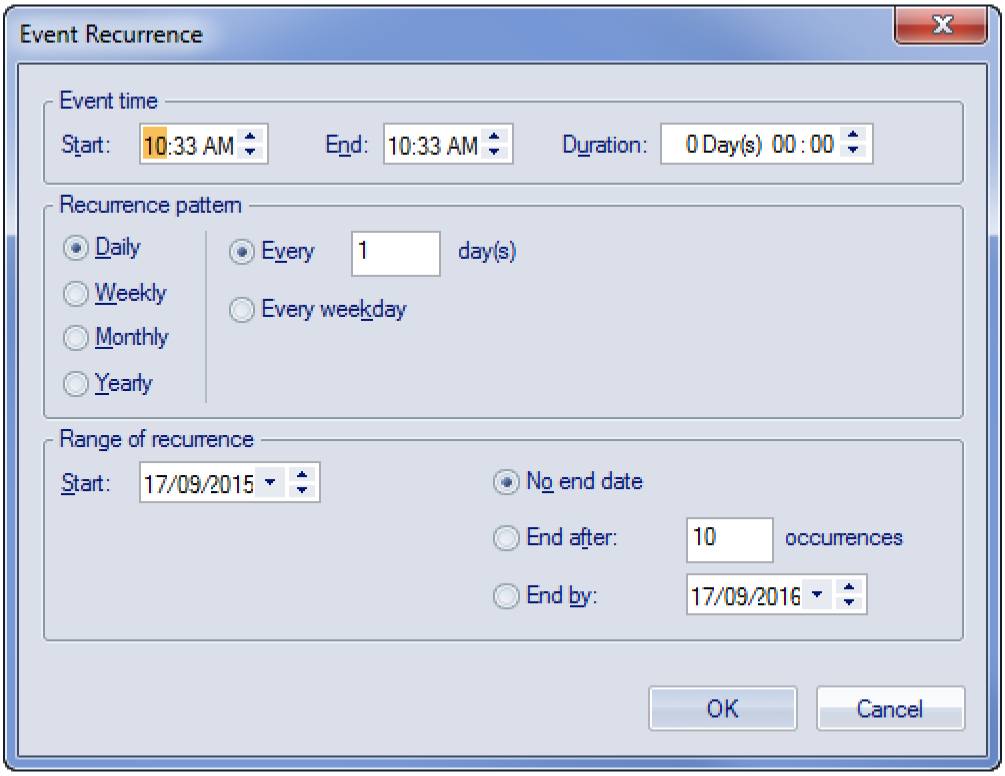
The Calendars Tab

This tab defines any number of special periods of time that can influence the process, such as working days, shifts, holidays or periodic events (for example, stock-taking, inventory or auditing). You select these calendar periods as factors affecting the progress of a particular action in the process; when you click on a calendar period on the 'Calendars' tab, the lower panel lists the activities and parameters that apply that period to the process.



To add a new calendar period

1. Click on the **New button** to display the 'Event Recurrence' dialog.



Event time

Start: 10:33 AM End: 10:33 AM Duration: 0 Day(s) 00 : 00

Recurrence pattern

● Daily
○ Weekly
○ Monthly
○ Yearly

● Every 1 day(s)
○ Every weekday

Range of recurrence

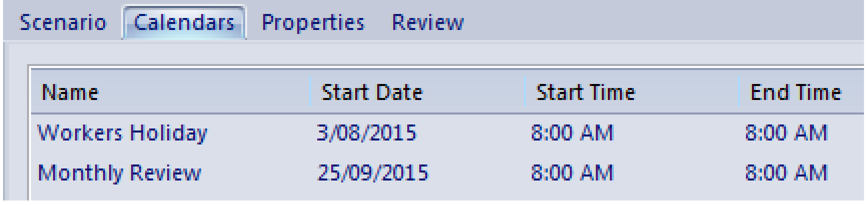
Start: 17/09/2015

● No end date
○ End after: 10 occurrences
○ End by: 17/09/2016

2. In the 'Event time' panel, the 'Start' and 'End' fields both default to the current time. The 'Start' field is the anchor; a change to either the 'End' field or the 'Duration' field automatically updates the other field, in reference to the 'Start' field.
   Click on the hour and minute segments of each field (and, for the 'Duration' field, the 'Day(s)' segment) separately, and use the 'spin' arrows to set the start time and the end time or **duration** of the period.

3. In the 'Recurrence pattern' panel, select the radio button for the interval at which the calendar period recurs. Each option displays an appropriate set of fields in the right of the panel for refining that interval to every day/week/month or every two/three/four days/weeks/months, on a particular day of the week, or day or date of the month, or day or date in the year. Select the checkboxes or values in drop-down lists as appropriate.
4. In the 'Range of recurrence' panel, select the date on which the calendar period takes effect and select the appropriate radio button to define when the period ceases to apply - never, after a set number of occurrences, or on a specific date. You can select an end date either from a drop-down calendar or using the 'spin' arrows on each segment of the date.
5. Click on the **OK button** to set the calendar period.

As you define calendar periods, they are listed in order of the start date and/or time, earliest first.



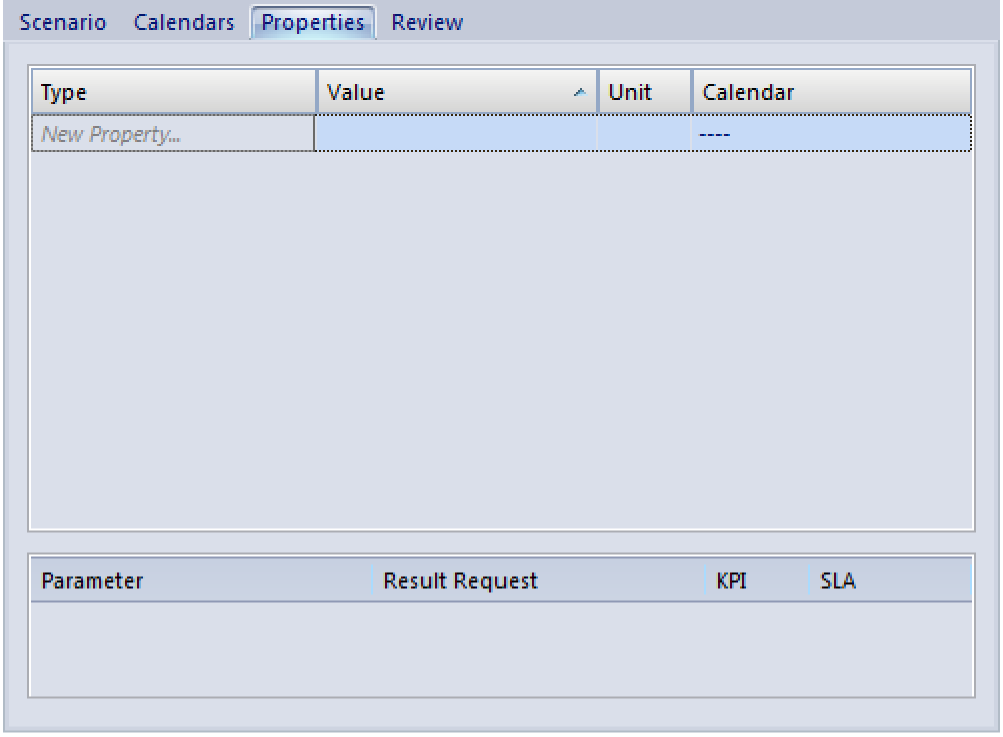
Name	Start Date	Start Time	End Time
Workers Holiday	3/08/2015	8:00 AM	8:00 AM
Monthly Review	25/09/2015	8:00 AM	8:00 AM

To edit a calendar period

Either click on the defined period and click on the **Edit button** to display the 'Event Recurrence' dialog again, or double-click on a value to change it directly on the 'Calendars' tab. All new calendar periods have the default name 'Calendar_n'; it is recommended that you directly edit the name to a string that identifies the nature or definition of the period.

The Properties Tab

This tab is used for defining **global** properties; that is, properties of the scenario, or properties that have even broader application, such as 'Number of Issues' or 'Net Cost'.



Type	Value	Unit	Calendar
New Property...			----

Parameter	Result Request	KPI	SLA

A number of objects throughout the configuration have their own 'Property' tab, for features specific to that object.

To create a global property

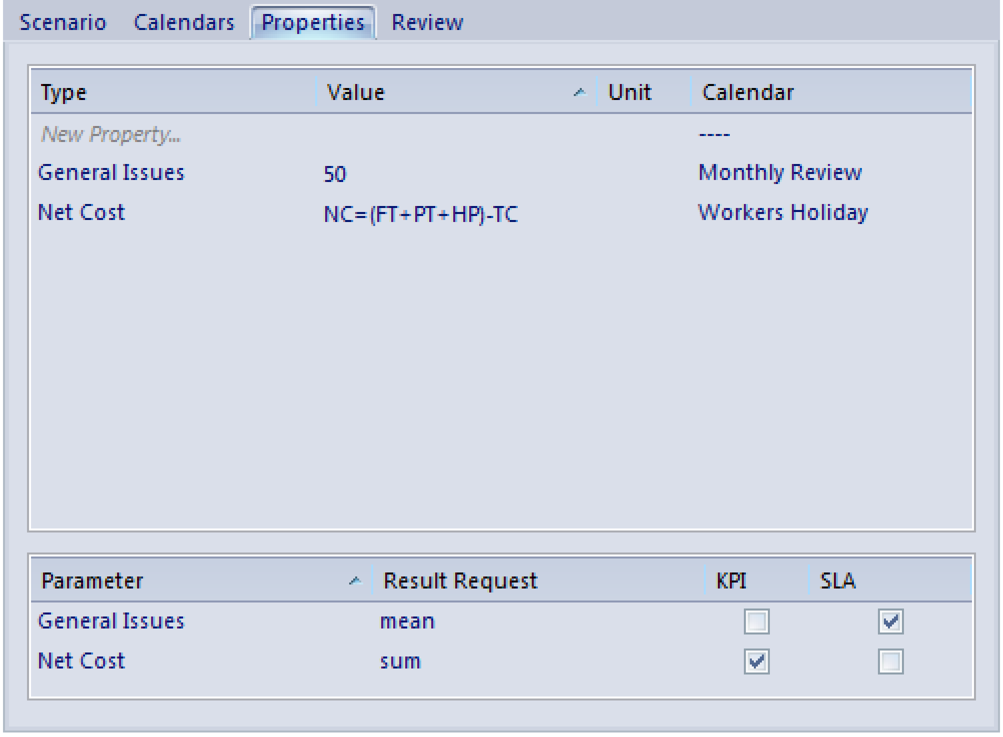
1. Click on the *New Property* text and type in the name of the property.
2. Click on the ... icon in the 'Value' field. The 'Parameter Value' dialog displays, from which you select the appropriate tab to define the value itself or the expression to derive the value. When you have defined the value, click on the **OK button**.
3. In the 'Unit' field, if necessary, type the unit of expression of the value you have defined. For example, Hours or $K.
4. In the '**Calendar**' field, if this property can be modified by a defined calendar period, click on the drop-down arrow and select the appropriate period.

Result Requests

In the lower panel of the 'Properties' tab, and of the object parameter tabs throughout the configuration, most of the possible types of parameter appropriate to the tab are listed. For many of these parameters, you can click on the drop-down arrow in the 'Result Request' column and select the type of processing of that parameter you want to check for in the simulation results, such as a count of the number of times it was used, or the sum, mean, minimum or maximum value used, in the context of a specific activity or of the Business Process as a whole. Certain parameters, such as 'Priority' and 'Probability', would not have any meaning in this context and either are not listed or cannot be set in the 'Result Request'.

You would then execute a **customized** simulation, which will only show the parameters you have selected to review.

You can also specify whether the parameter is a 'Key Performance Indicator' (KPI) or part of a 'Service Level Agreement' (SLA), by selecting the appropriate checkbox against the parameter.



Type	Value	Unit	Calendar
New Property...			----
General Issues	50		Monthly Review
Net Cost	NC=(FT+PT+HP)-TC		Workers Holiday

Parameter	Result Request	KPI	SLA
General Issues	mean	□	☑
Net Cost	sum	☑	□

The Review Tab

The 'Review' tab presents a summary of the saved changes that have been made to the object parameters and properties in the configuration, during the current work session. You cannot change the data captured on this tab.

BPMN Validation

It is possible to validate your BPMN Model and BPSim Configuration. Clicking the **Validate button** in the configuration will validate the model and configuration; all errors and warnings will be reported in the **System Output** window.

Alternatively, you can select a root Package containing a BPMN model and BPSim Configuration, and then select the 'Package | **Model Validation** | **Validate** BPMN Model in Current Package' option.

The BPSim Control Perspective

The Control Perspective targets how activity flows through the process, moderated by the likelihood of a sequence of events and the priorities of certain events when they occur.

When you initially select 'Control' in the 'Perspective' field, the hierarchy on the left of the screen groups all elements in the Business Process (including the Business Process itself) by type and, for each element that has one or more connections to other elements, lists those connections by name and/or by the name of the target element. The 'Configure BPSim' screen also initially displays the 'Trigger Count' tab.

Access

Use one of the methods outlined in the table below, to open the 'Configure BPSim' window and load a BPSim Artifact element, using the 'Control' perspective.

Ribbon	Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ≡ > Select Artifact > [select a BPSim Artifact] > Set 'Perspective' to 'Control' Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ≡ > [select a package] > Create Artifact : Add New > [provide a name] > Set 'Perspective' to 'Control' Simulate > BPSim > Manage > Find BPSim Configuration Artifacts > double-click an Artifact in search results > Set 'Perspective' to 'Control'
Context Menu	In **Project Browser** or Diagram | Right-click BPSim Artifact element | Configure BPSim | Set 'Perspective' to 'Control'

The Control Tab

The 'Control' tab displays when you click on the name of a Start Event, Intermediate Event or connector in the hierarchy. Through this tab you can define a small number of parameters, dependent on the nature of the object; some objects cannot take control parameters, some can take only certain types, and others a wider range.

To define a control parameter, click on the drop-down arrow after the *New Parameter* text in the 'Type' column and select from the list of available parameters. An object that cannot take parameters has the text *No parameter items* in the 'Type' column.

If you have set any applicable calendar periods, for each parameter you can click on the drop-down arrow in the '**Calendar**' field and select the period during which the parameter takes effect. You can set multiple parameters of one type if the value varies according to time or chronological event.

The Trigger Count Tab

The 'Trigger Count' tab lists the elements in the Business Process, grouped by stereotype. Against each element, in the 'Trigger Count' column, is a checkbox that you can select to capture the number of times the element is accessed during a simulation. It provides a quick and simple way to specify which actions to specifically monitor in a customized simulation. You can also use the 'Result Request' column in the 'Properties' or 'Control' tabs to capture the count of accesses for each element individually (and those settings update and are updated by the 'Trigger Count' tab), but the tab helps you set or clear the count requests in one place. You do need to use the 'Result Request' column to capture the number of times a pathway (connector) *between* elements is accessed.

On the 'Trigger Count' tab you can, if you prefer, use the arrows in the column heads and/or the **Filter bar** to list the elements in a different sequence by Stereotype, Name or selected/cleared 'Trigger Count' checkbox.

Control Parameters

Parameter	Description
Condition	This parameter simply defines some type of condition that must be satisfied before processing continues. It can be either a simple constant, a statistically-derived value, a value calculated from an expression, or an enumerated value. In the 'Value' field, click on the ... button and work on the 'Value' dialog.
Probability	This parameter assigns a weighting to the object or connector so that when there is a choice of paths, the simulation will process that object or connector a fixed proportion of the times. Probability is expressed as a decimal portion of 1.0, so in the 'Value' field you type a value such as 0.25 (25% of the time, the simulation will process this object). You should check that the probabilities of all paths in the same choice add up to 1.0.
InterTriggerTimer	This parameter defines the time interval between triggers for the action that the object or connector represents. This interval can be either a simple constant, a statistically-derived value, a value calculated from an expression, or an enumerated value. In the 'Value' field, click on the ... button and work on the value dialog as for a global property.
TriggerCount	The TriggerCount sets the number of times the object or connector should be processed during one execution of the simulation, so that the processing cycle imitates the typical demand for a particular chain of actions. It is generally set for Start Event elements. Note that this TriggerCount **sets** the number of times the element or connector **is to be** processed in a normal or customized simulation, whilst the TriggerCount in the 'Trigger Count' tab and 'Result Requests' **gets** the number of times the element or connector **has been** processed in a customized simulation.

The Priority Tab

The 'Priority' tab displays when you select an Activity element. The tab identifies whether the activity can be interrupted by another activity, and what the relative priority is of the activity if there is a choice as to which activity is executed first.

Parameter	Description
Interruptable	This parameter defines whether the activity can be interrupted, and takes a value of **True** or **False**. You can create more than one instance with different values for different calendar periods.
Priority	This parameter defines the priority of the activity relative to other activities that

	might be executed at the same time. In the 'Value' field, click on the ... button and work on the value dialog to set either a simple constant or a derived value for the priority. Take into consideration whether the competing activities are interruptable and what their priority values are. You can set different priorities for the same activity, corresponding to different calendar periods.

The BPSim Resource Perspective

The 'Resource Perspective' targets the involvement of types and roles of workers and other resources in a process, reviewing the required numbers, combinations, costs and availability. Before you define how the resources can be deployed in the process, you must create the appropriate BPMN 2.0 Resource elements within your model. These are then listed in the 'Resource' folder in the 'Configure BPSim' hierarchy. You can also use ResourceRoles that you might have defined in your model, but it is quicker and easier to create and use any roles you require within the BPSim configuration.

Access

Use one of the methods outlined in the table below, to open the 'Configure BPSim' window and load a BPSim Artifact element, using the 'Resource' perspective.

Ribbon	Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > Select Artifact > [select a BPSim Artifact] > Set 'Perspective' to 'Resource' Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > [select a package] > Create Artifact : Add New > [provide a name] > Set 'Perspective' to 'Resource' Simulate > BPSim > Manage > Find BPSim Configuration Artifacts > double-click an Artifact in search results > Set 'Perspective' to 'Resource'
Context Menu	In **Project Browser** or Diagram | Right-click BPSim Artifact element | Configure BPSim | Set 'Perspective' to 'Resource'

The Resource Tab

The 'Resource' tab is available when you click on a Resource element in the 'Resource' folder. On this tab, you define how many instances of each resource are available for each calendar period that you have defined, and any roles the resource might be specialized into. You can provide multiple instances of each resource parameter, depending on how the resource availability varies across any defined calendar periods.

When you have finished defining the resource availability, click on the **Save button**. The 'Resource' tab has no 'Result Request' facility.

Parameter	Description
Availability	In the 'Type' column, select this parameter. In the 'Value' column, click on the drop-down arrow and select either '**True**' (the resource is available) or '**False**' (this resource is not available). You can also define mechanisms for establishing availability, using the 'Parameter Value' dialog, as long as they result in a value of True or False. In the '**Calendar**' column, click on the drop-down arrow and select the calendar period to which this availability applies.
Quantity	In the 'Type' column, click on the drop-down arrow and select this parameter. In the 'Value' column, either type in a simple numerical value or click on the

	drop-down arrow and use the 'Parameter Value' dialog to define a constant or derived value for the quantity of this resource that is available for assignment. In the '**Calendar**' column, click on the drop-down arrow and select the calendar period during which this quantity of the resource is available.
Role	To add a role to the panel, click on the *New Role* text and type in the role name. At this point the new roles remain as parameters within the BPSim configuration and are not reproduced in the model.

The Cost Tab

The 'Cost' tab is available when you select an Activity element or a Resource element in the 'Configure BPSim' hierarchy. To add a parameter, click on the drop-down arrow after the 'New Parameter' text in the 'Type' column. You can have multiple costs if the cost varies according to the calendar period in which it is incurred.

Parameter	Description
Fixed Cost	This parameter defines the fixed cost of calling the resource or activity into use, automatically expressed in the unit of currency defined on the 'Scenario' tab. For example, the call-out fee for an electrician. In the 'Value' column, either type in the cost or click on the drop-down arrow and define how the cost is derived. A derived cost must resolve to a floating or numeric value. If necessary, in the '**Calendar**' column click on the drop-down arrow and select the calendar period during which the fixed cost applies.
Unit Cost	This parameter defines the unit cost of using the resource or running the activity, automatically expressed in the unit of currency and per the unit of time defined on the 'Scenario' tab. For example, the hourly rate of the electrician. In the 'Value' column, either type in the cost or click on the drop-down arrow and define how the cost is derived. A derived cost must resolve to a floating or numeric value. If necessary, in the '**Calendar**' column click on the drop-down arrow and select the calendar period during which the unit cost applies.

The ResourceRole Tab

The 'ResourceRole' tab is available when you select an Activity element or Business Process element in the 'BPSim Configuration' hierarchy. Its main purpose is to define the resources for an Activity, but if necessary you can assign resources that act within the Business Process definition itself.

You can apply the resources as defined within the model. Alternatively, you can define a combination of resources and roles that is not identified in the model, which overrides the original definition. For example, the model might show that an Activity 'Take Customer Call' is associated with the resource 'Front Office'. In the BPSim configuration you can qualify that relationship to indicate that the Activity actually uses 50 Front Office staff. But you might instead override the model relationship to indicate that the work can be done by 50 resources from either Front Office or Support, if (for example) the Support resources have the role of Switchboard Operators.

The resource selection for the Activity is also displayed as an expression, at the bottom of the tab.

Panel	Description

Resource Roles defined in BPMN	In this panel, you assign a resource to the selected Activity in the BPMN model, as shown: 1. Click on the **Add button**, which adds a ResourceRole element to the 'Type' column and to the Activity in the model. 2. Click on the 'BPMN Resource' column and on the ... button, and select a Resource from the **Tagged Values** browser (which assigns the resource to the 'ResourceRef' **Tagged Value** of the new ResourceRole element). 3. If necessary, type a role for the resource in the 'Role(Name)' column (preferably using the roles you defined for the resource on the 'Resource' tab, above). This value is reflected in the ResourceRole element defined for this Activity. 4. In the 'Quantity' column, the quantity of this resource that the Activity uses on each iteration defaults to 1. If necessary, overtype this number with a different value. 5. Click on the **Save button**. Note that the quantity is reflected in the 'Advanced Override for Single Resource Role' panel. 6. If you need to add another type of resource, or a different role of the same resource, you can repeat steps 1 to 5 **or** (preferred) use the Advanced Override. Note that the quantity reflected in the 'Advanced Override for Single Resource Role' panel is for one selected ResourceRole entry.
Advanced Override for Single Resource Role	In this panel, you can define a *combination* of resources to use in place of a relationship defined in the model. All resources currently defined in the model are automatically listed in the panel. The override applies to a single selected ResourceRole entry in the 'Resource Roles defined in BPMN' panel; if you have more than one entry in this panel, set the 'Quantity' value for the other entries to 0. 14. Select the radio button to indicate whether the combination of resources you are defining is an AND relationship (so many of Resource X and so many of Resource Y at the same time), or an OR relationship (0 to n of Resource X OR 0 to n of Resource Y at any time). 15. Double-click in the 'Quantity Required' column and type the required number of each resource for the activity. 16. Click on the **Save button**. Note that the 'Resource Selection Expression for Activity' panel at the foot of the tab shows the expression that will apply this combination of resources. If a resource is defined in the 'Resource Roles defined in BPMN' panel, a change to the quantity of that resource is reflected in that panel, but the addition of a subsequent resource in the 'Advanced Override for Single Resource Role' panel changes the value of the 'Quantity' field to 'Override'. If you want to make the **role** of a resource significant in the combination, you must have at least one resource identified in the upper panel. This adds a 'Role' item to the end of the list in the 'Advanced Override for Single Resource Role' panel. 1. Click on the 'Role' item and on the ... icon in the 'Resource or Role' column. The 'Roles for Selection' dialog displays, listing the roles available for each resource in the configuration (as you defined on the 'Resource' tab, previously). 2. Select the checkbox against the required role (or more than one role, if appropriate) and click on the Save button. The roles are shown in the 'Resource or Role' column. 3. In the 'Quantity Required' column, type in the required number of resources of this role or these roles. 4. Click on the Save button. The 'Role' item moves underneath the appropriate resource name, and another empty 'Role' item is added to the end of the list. The 'Resource Selection Expression for Activity' panel is updated to show the

	resource roles you have specified. If you do not want to keep a resource or role in the defined combination, set its 'Quantity Required' value to 0. It will remain at this value while you remain on this Activity, but when you return to the Activity after working elsewhere the field is clear.

Learn more

- Business Process Simulation (BPSim)
- Working in BPSim
- The BPSim Control Perspective
- The BPSim Time Perspective
- Configuration Inheritance
- BPSim Charts

The BPSim Time Perspective

The 'Time' (or 'Temporal') Perspective helps you to examine how the **duration** of one or more phases in the processing of an Activity influences the business process. In this Perspective you only set parameters on Activity elements, but you can monitor (through the 'Result Request' column) the occurrence, minimum, maximum, mean and sum duration of a parameter for the Business Process as a whole or for the Activities involving a defined Resource.

Access

Use one of the methods outlined in the table below, to open the 'Configure BPSim' window and load a BPSim Artifact element, using the 'Temporal' perspective.

Ribbon	Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > Select Artifact > [select a BPSim Artifact] > Set 'Perspective' to 'Temporal' Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > [select a package] > Create Artifact : Add New > [provide a name] > Set 'Perspective' to 'Temporal' Simulate > BPSim > Manage > Find BPSim Configuration Artifacts > double-click an Artifact in search results > Set 'Perspective' to 'Temporal'
Context Menu	In **Project Browser** or Diagram | Right-click BPSim Artifact element | Configure BPSim | Set 'Perspective' to 'Temporal'

The Time Tab

On this tab, for a selected Activity element, you define the **duration** of a number of stages through which the Activity might pass. For each stage you:

- Click on the drop-down arrow in the 'Type' field and select the type as a parameter
- Click on the ... button in the 'Value' field and define the duration (either as a fixed value or as a derived value) and unit of measurement, and
- If appropriate, click on the drop-down arrow in the '**Calendar**' field and select a calendar period to which this duration applies

Parameter	Description
Processing Time	The time spent actually performing the Activity. This can be strictly defined, or more broadly set to represent the **duration** of the Activity (the duration being the time it takes to complete the Activity; in strict use, the sum of the 'Setup Time', 'Processing Time', 'Validation Time' and 'Rework Time'). For this and the other six parameters, you use the 'Parameter Value' dialog to define a constant or derived value in the appropriate unit of time; a derived value must resolve to a NumericParameter, a FloatingParameter or a DurationParameter.
Queue Time	The delay between the Activity being offered and the Activity being processed; a refinement of the **lag time**.

Rework Time	The time spent redoing or correcting the work of the Activity; a refinement of the **duration**.
Setup Time	The time expended between the element being offered and the element being allocated to perform the actual work; a refinement of the **duration**.
Transfer Time	The time spent traveling from the previous processing step; a refinement of the **lag time**.
Validation Time	The time taken to review or inspect the work done; a refinement of the **duration**.
Wait Time	The time between the element being allocated and its action actually starting. This can be strictly defined, or more broadly used to represent the **lag time** of the Activity (the lag time being the time between the action of one Activity stopping and the action of the next Activity starting; in strict use the sum of the 'Transfer Time', 'Queue Time' and 'Wait Time').

Notes

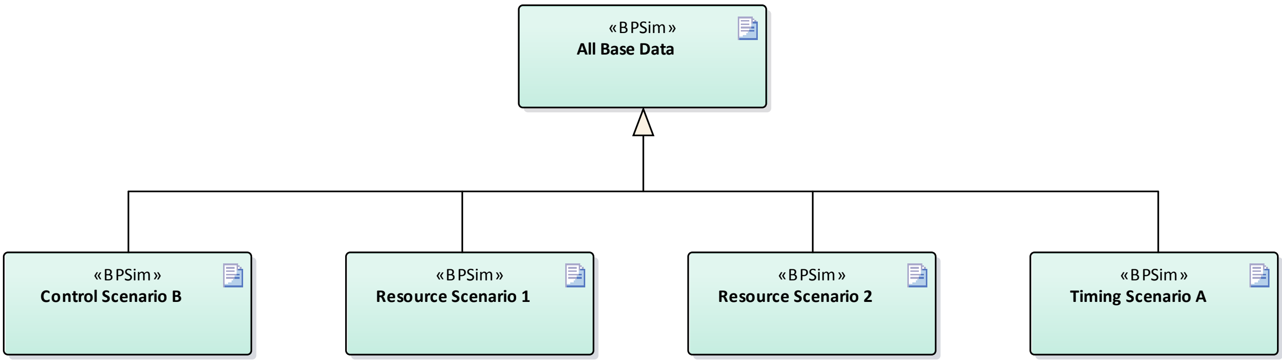
- In the BPSim configuration you can set values for seven different time parameters; however, the BPSim Simulation engine, during a simulation, amalgamates the values of these parameters into one ProcessingTime parameter
  You can vary the initial values and value generation methods of the individual parameters to observe the effect each has on the eventual processing time, or to define a simulation to be exported into a different tool for analysis and/or execution
- An example of the behavior of the Time parameters during simulation is provided in the EAExample model

Configuration Inheritance

When you define a BPSim configuration it can contain a lot of data, some relating to each of the three separate **Perspectives**, and some being base data for the configuration grouped on three pages or tabs of the 'Configuration' dialog. A major objective of performing simulations of a business process using BPSim is to observe the effects of varying the values of certain parameters, whilst leaving others unchanged. It would be very useful, then, to maintain all of the standard data in a single place where it is created only once, and work on the variable data in one or more separate places that refer back to the base data. This would also mean that if you changed any of the standard data, the changes would be reflected in all the configurations that refer to it.

In BPSim, you can achieve this arrangement using inheritance between configuration Artifact elements.

In essence, you would define the standard data in a base configuration, and the data you are manipulating (and only that data) in configurations called, say, Resource Scenario 1, Resource Scenario 2, Timing Scenario A, or Control Scenario B. You would then create Generalize connectors from the scenario configuration Artifacts (the sources) to the base configuration Artifact (the target).



When you run a simulation on a scenario configuration, it will inherit the data defined in the base configuration and the simulation will perform as if it were processing a single configuration containing all the data. If the scenario configuration contains data that is also defined in the base, the (changed) values in the scenario take precedence.

Comparisons

To keep track of how the base and scenario configurations have been set up, and what the differences are between them, you would run a 'BPMN Simulation Configuration' report on one of them and drag the other(s) into the same report.

Whether or not you select the 'Show Only Different Items' option or the 'Highlight Different Items' option, you will see all the base configuration data in one column and a much smaller volume of scenario data in another. Where the data occurs in both columns, you can compare values and check what the scenario values have been changed to.

Similarly, when you run simulations on both configurations, you can display the 'BPMN Simulation **Report View**' and compare the results of the simulations. In this case if you select either the 'Show Only Different Items' or the 'Highlight Different Items' option, you will see a much smaller data set where the scenario values differ from the base configuration values, because in the **simulation** the scenario will have inherited all of the base configuration data and will produce substantially the same output.

Using the MDG BPSim Execution Engine

The MDG BPSim Execution Engine is a separate **Add-In** that you can purchase to execute the simulations that you have defined using the Business Process Simulation (BPSim) facility in Enterprise Architect.

Install the BPSim Execution Engine

You can purchase and download the MDG BPSim Execution Engine from the 'Products' page of the Sparx Systems website (www.sparxsystems.com/products). As part of this process, you will receive a registration key by email, to enable you to use the system beyond a 14-day trial period.

The MDG BPSim Execution Engine installer does not include JRE or JDK, therefore you must install JRE version 1.7 or higher to use the simulation and, if your BPSim Configuration contains any property parameters, you must also install JDK version 1.7 or higher.

You do not need to do any further configuration of the engine unless you have multiple versions of JRE/JDK on your system and you want to specify which version the execution engine should use. In this case, apply these environment variables as shown:

1. Click on the Windows 'Start' icon and select the 'Computer' option.
2. From the banner menu, select the 'System properties' option.
3. From the side panel, select the 'Advanced system settings' option.
4. On the 'Advanced' tab of the 'System Properties' dialog, click on the **Environment Variables button**.
5. On the 'Environment Variables' dialog, in the 'System variables' panel, click on the **New button**.
6. On the 'New System Variable' dialog, complete the fields with the values shown:

   For JRE: Variable name: MDG_BPSIM_JRE_HOME
   Variable value: C:\Program Files\Java\jre7

   For JDK: Variable name: MDG_BPSIM_JDK_HOME
   Variable value: C:\Program Files\Java\jdk1.7.0_51
7. Click on the **OK button**.
8. You must re-start your machine for the new variables to take effect.

Access

Open the 'Configure BPSim' window, load a BPSim Artifact and click the **Start icon** ▶ on the toolbar.

Ribbon	Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > Select Artifact > [select a BPSim Artifact] > click the **Start icon** ▶ Simulate > BPSim > Manage > Open BPSim Manager > click toolbar icon ☰ > [select a package] > Create Artifact : Add New > [provide a name] > click the Start icon ▶ Simulate > BPSim > Manage > Find BPSim Configuration Artifacts > double-click an Artifact in search results > click the Start icon ▶

Context Menu	Right-click BPSim Artifact element | Configure BPSim > ▶
Other	Right-click BPSim Artifact element | Simulate BPMN with BPSim

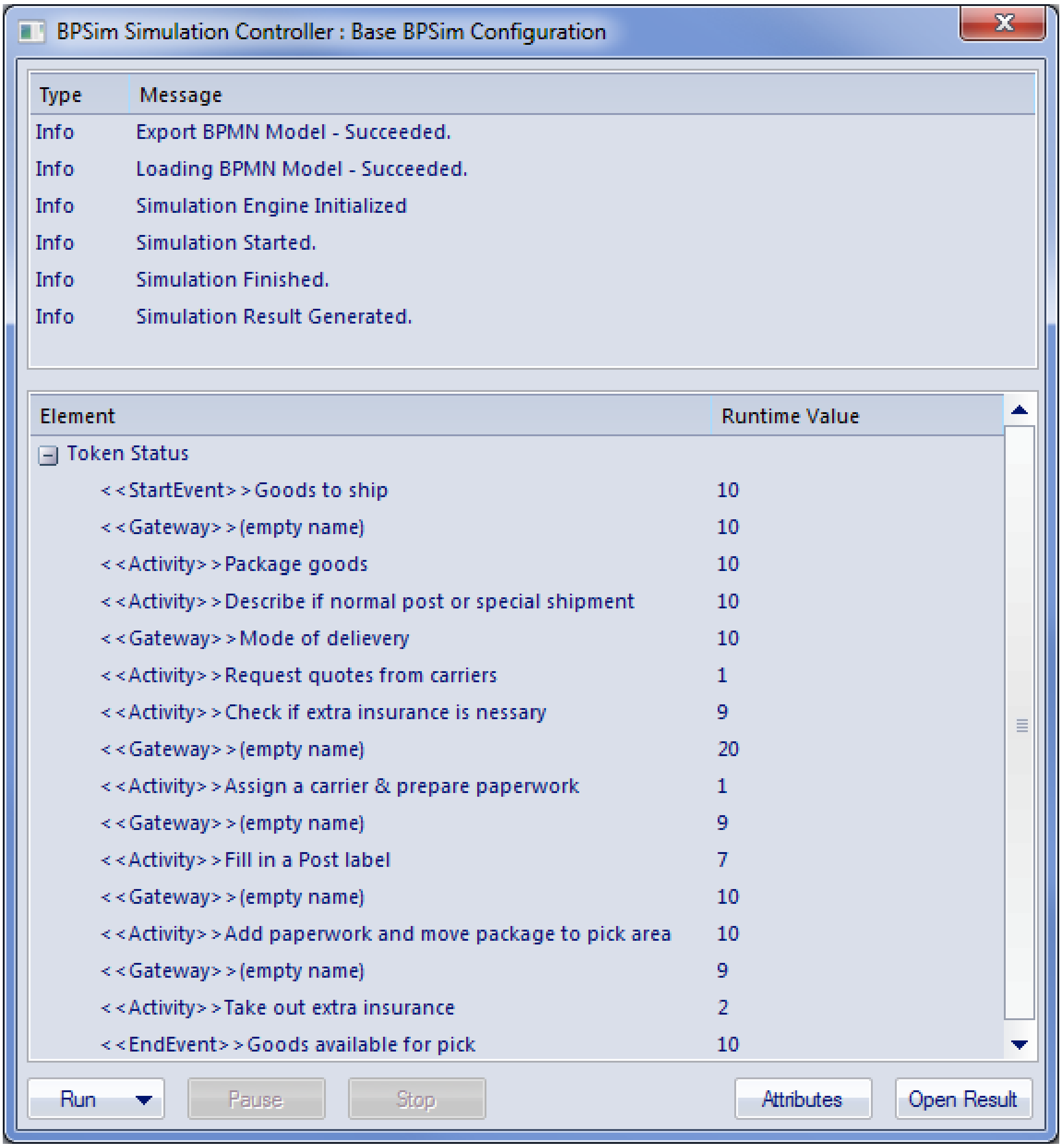
Execute and Control a Simulation

Click on the **Run button** and select either:

- 'Standard Simulation' or
- 'Customized Simulation'

These options trigger the same processing, except that while 'Standard Simulation' generates a report on all built-in parameters set in the simulation, 'Customized Simulation' extracts results for only the parameters you have specifically flagged using the 'Result Request' columns in the configuration.

The simulation executes, displaying processing messages in the top section of the dialog, and the elements and parameters processed with the runtime values used from the configuration.



Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>Goods to ship	10
<<Gateway>>(empty name)	10
<<Activity>>Package goods	10
<<Activity>>Describe if normal post or special shipment	10
<<Gateway>>Mode of delievery	10
<<Activity>>Request quotes from carriers	1
<<Activity>>Check if extra insurance is nessary	9
<<Gateway>>(empty name)	20
<<Activity>>Assign a carrier & prepare paperwork	1
<<Gateway>>(empty name)	9
<<Activity>>Fill in a Post label	7
<<Gateway>>(empty name)	10
<<Activity>>Add paperwork and move package to pick area	10
<<Gateway>>(empty name)	9
<<Activity>>Take out extra insurance	2
<<EndEvent>>Goods available for pick	10

While the simulation is in progress, you can use the **Pause button** and **Resume button** to temporarily halt processing and then resume, or you can click on the **Stop button** to cancel the simulation altogether.

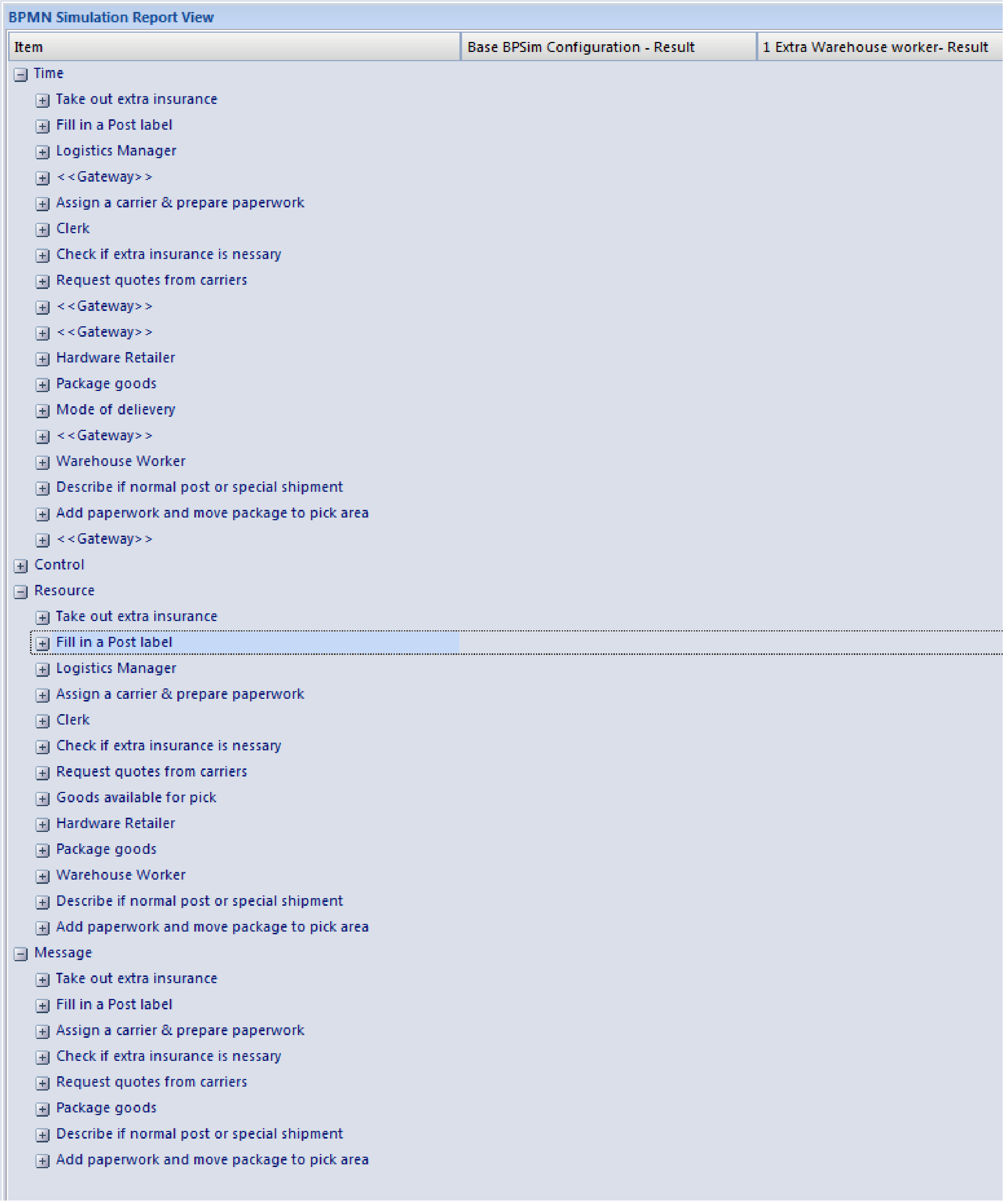
The results of the simulation are written to an Artifact element added to the Business Process parent Package. A Standard simulation writes to a <<BPSimReport>>-stereotyped Artifact, whilst a Customized simulation writes to a <<BPSimCustomReport>>-stereotyped Artifact.

Track Property Values

As well as the built-in parameters, you can define your own process-specific Property parameters (attributes) in the configuration. When the simulation has finished, and if you have defined property parameters, the **Attributes button** is enabled. When you click on this button the 'BPSim PropertyParameter Values' dialog displays, through which you can track how the run-time values of all the property parameters accrue or change through the business process.

Review a Simulation

When the simulation has finished processing, click on the **Open Result button**. The 'BPMN Simulation **Report View**' tab opens in the main work area, showing the results for the built-in parameters in the current simulation (but not for the user-defined property parameters). If you have already run a simulation of another configuration based on the same business process, that is also displayed in the report as an additional column. Otherwise, you can click on the report Artifact element and drag it onto the report tab, to compare the runtime values of the built-in parameters under two (or more) configurations.



BPMN Simulation Report View

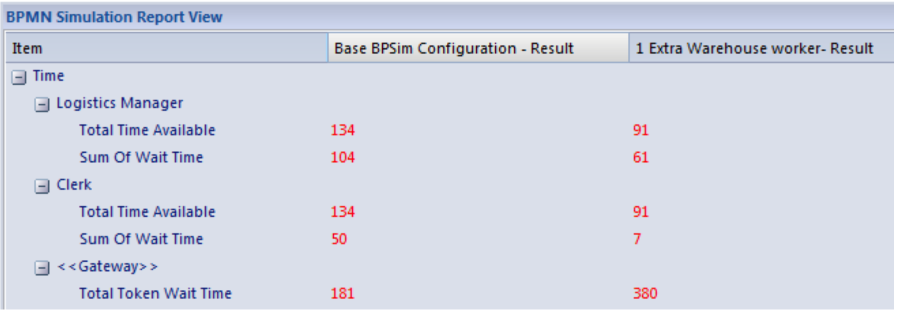
Item	Base BPSim Configuration - Result	1 Extra Warehouse worker- Result
⊟ Time		
⊞ Take out extra insurance		
⊞ Fill in a Post label		
⊞ Logistics Manager		
⊞ <<Gateway>>		
⊞ Assign a carrier & prepare paperwork		
⊞ Clerk		
⊞ Check if extra insurance is nessary		
⊞ Request quotes from carriers		
⊞ <<Gateway>>		
⊞ <<Gateway>>		
⊞ Hardware Retailer		
⊞ Package goods		
⊞ Mode of delievery		
⊞ <<Gateway>>		
⊞ Warehouse Worker		
⊞ Describe if normal post or special shipment		
⊞ Add paperwork and move package to pick area		
⊞ <<Gateway>>		
⊞ Control		
⊟ Resource		
⊞ Take out extra insurance		
⊞ Fill in a Post label		
⊞ Logistics Manager		
⊞ Assign a carrier & prepare paperwork		
⊞ Clerk		
⊞ Check if extra insurance is nessary		
⊞ Request quotes from carriers		
⊞ Goods available for pick		
⊞ Hardware Retailer		
⊞ Package goods		
⊞ Warehouse Worker		
⊞ Describe if normal post or special shipment		
⊞ Add paperwork and move package to pick area		
⊟ Message		
⊞ Take out extra insurance		
⊞ Fill in a Post label		
⊞ Assign a carrier & prepare paperwork		
⊞ Check if extra insurance is nessary		
⊞ Request quotes from carriers		
⊞ Package goods		
⊞ Describe if normal post or special shipment		
⊞ Add paperwork and move package to pick area		

To make it easier to view the data from the report, you can drag the 'BPMN Simulation Report View' tab out of the main view so that it becomes a floating window, and enlarge the window to a suitable size.

Click on the expansion boxes against the parameters you want to check. You can also expose and filter the information using right-click context menu options.

You can represent specific differences between the results from separate simulations as Charts. The simulation result Artifacts (<<BPSimReport>>*name* - Result elements) must exist before you can set up the Chart Artifacts. There is a template Chart Artifact for Standard simulations and one for Customized simulations.

BPMN Simulation Report Options

Option	Description
Collapse All	Select this option to collapse the parameter hierarchy down to just the parent tab names.
Expand All	Select this option to expand the parameter hierarchy out to the lowest value type.
Show Only Different items	(When you have two or more simulations shown.) Select this option to restrict the display to those parameters where the values differ between the simulations. Click on the option again to deselect it.
Highlight Different Items	(When you have two or more simulations shown and where some of their parameter values are different.) Displays the differing parameter values in red. This option is disabled if you select the 'Show Only Different Items' option.  BPMN Simulation Report View Item | Base BPSim Configuration - Result | 1 Extra Warehouse worker- Result ⊟ Time ⊟ Logistics Manager Total Time Available | 134 | 91 Sum Of Wait Time | 104 | 61 ⊟ Clerk Total Time Available | 134 | 91 Sum Of Wait Time | 50 | 7 ⊟ <<Gateway>> Total Token Wait Time | 181 | 380
Show Only Non Empty Items	Select this option to filter the display to show only parameters that have a specific value other than 0.
Remove Model	(When you have selected a specific result, which identifies the simulation in the report.) Select this option to remove the simulation column from the report.

BPSim Execution Engine - Simulation language, expression and property parameters

The MDG BPSim Execution Engine supports simulation on XPath 1.0 or Java, where the appropriate language is defined as the Expression Language in the simulation configuration. It also supports the use of process instance data in BPSim Property Parameters, where the actual value is only determined during execution.

XPath 1.0 Operators

These operators can be used in BPSim expression parameters.

Operator	Description
|	The Union operator, used for resource acquisition. Example: getResource(('w1',1) | getResource('w2',1)
+	Addition. Example: 4 + 6
-	Subtraction. Example: 6 - 4
*	Multiplication. Example: 6 * 4
div	Division. Example: 8 div 4
=	Equality. Example: 4 = 4 (**True**)
!=	Not Equal. Example: 5 != 3
<	Less than. Example: 6 < 9
<=	Less than or equal to. Example: x <= 6
>	Greater than. Example: 9 > 6
>=	Greater than or equal to. Example: n >+ 7
or	Alternative. Example: n = 6 or n <= 6

and	Combination. Example: n = 5 and m < 8
mod	Modulus division. Example: 5 mod 2
getProperty	Get a property value. Example: getProperty ("amount")
getResource	Get a resource assignment. Example: getResource ('w1',1)

Note

The Expression Language can be set in the 'Configure BPSim' dialog, on the 'Scenario' tab; the two options 'XPath 1.0' and 'Java' are available.

If you select 'Java', you must set the property 'JDK Home' to a valid JDK directory.

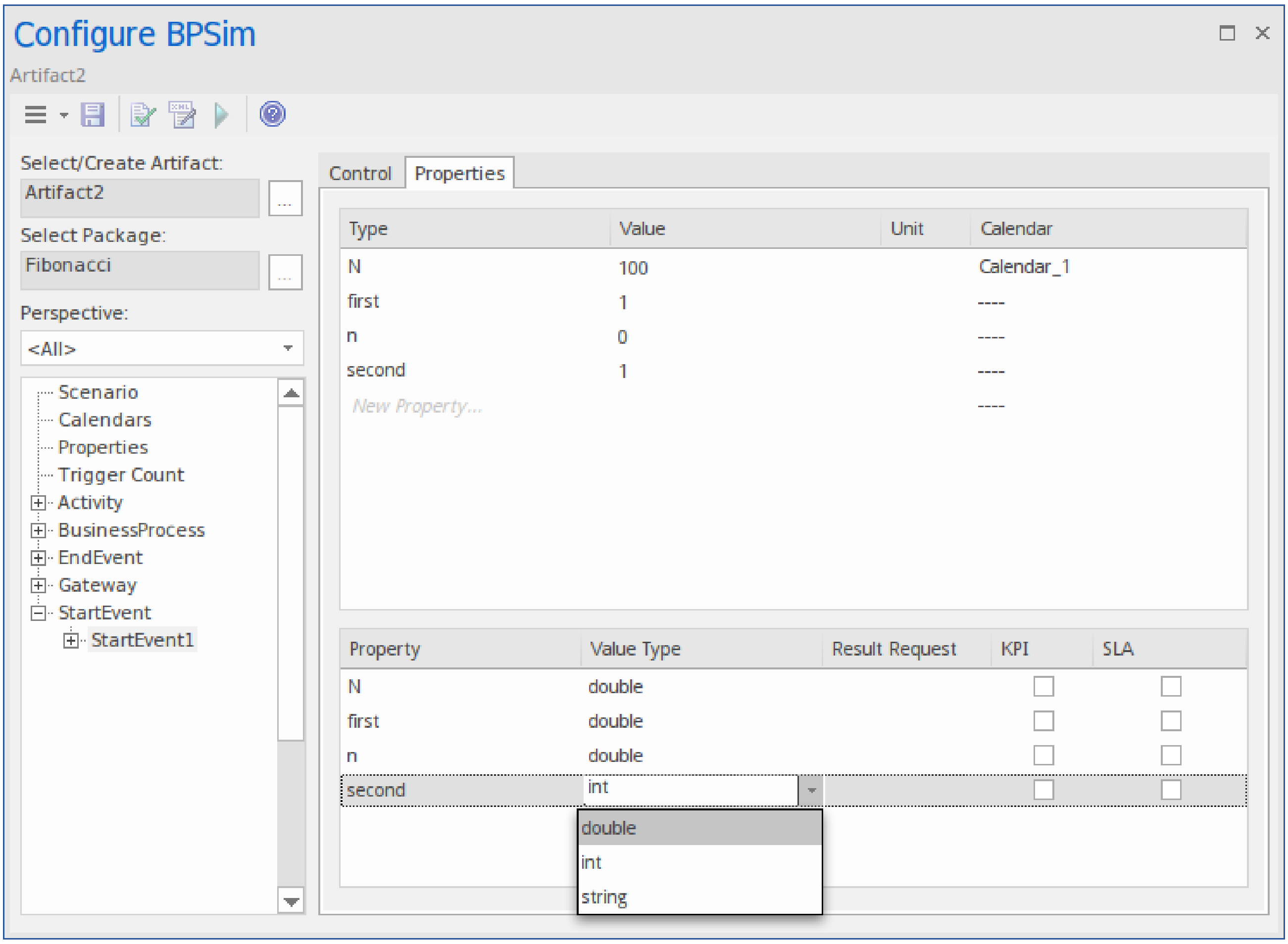
Tip: You can use {PropertyParameterName} as a short form of getProperty('PropertyParameterName'), which is useful when writing the value for the expressions; for example:

{n} < {N} instead of getProperty('n') < getProperty('N')

The short form of the getProperty operator can be used in both XPath 1.0 and Java.

BPSim Property Parameters

In Enterprise Architect release 13.0 and higher, BPSim property parameters can hold process instance data to which no value is assigned until run time. You can define the property parameter type on the 'Properties' page of the 'Configure BPSim' dialog; the supported types are 'int', 'double' and 'string'.



Configure BPSim

Artifact2

Select/Create Artifact: Artifact2

Select Package: Fibonacci

Perspective: <All>

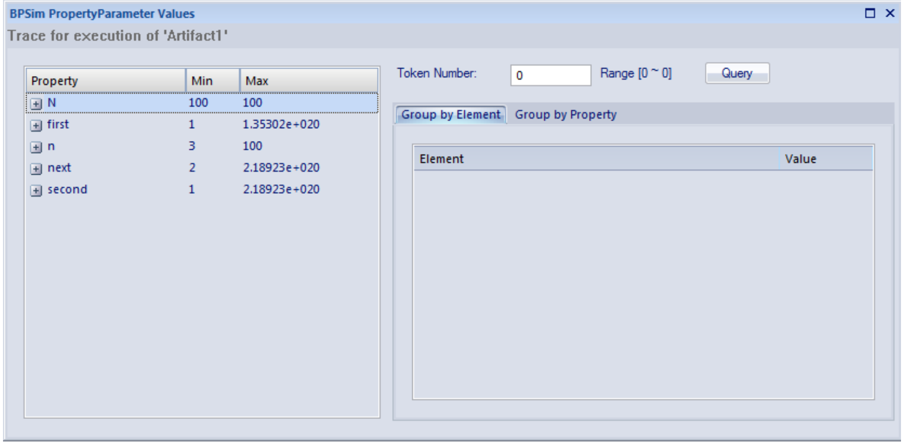
Type	Value	Unit	Calendar
N	100		Calendar_1
first	1		----
n	0		----
second	1		----
New Property...			----

Property	Value Type	Result Request	KPI	SLA
N	double		□	□
first	double		□	□
n	double		□	□
second	int		□	□

Tracking Property Parameter Values

The Business Process Simulator (BPSim) helps you to model and test the operating details of a business process, such as the resourcing of activities and tasks, the intervention of events, and the impact of decision points and the decisions made at those points. You add these process-specific property parameters, or attributes, to the BPSim configuration and, as you run simulations on the model according to the configuration, the BPSim engine helps you to capture the run-time values of the property parameters for every iteration of the simulation, and to filter the results to examine specific pathways or decision points. This gives you an incredibly detailed insight into what might actually happen in your business process under a specific condition or combination of conditions, either to generate a result or to show what processing path produces that result.

Access

Context Menu	Right-click on a defined BPSim Configuration Artifact |Simulate BPMN with BPSim... : Run (select simulation type) : Attributes (The **Attributes button** is not available if the simulation configuration does not contain any property parameters)  BPSim PropertyParameter Values Trace for execution of 'Artifact1' Property | Min | Max ⊞ N | 100 | 100 ⊞ first | 1 | 1.35302e+020 ⊞ n | 3 | 100 ⊞ next | 2 | 2.18923e+020 ⊞ second | 1 | 2.18923e+020 Token Number: 0 Range [0 ~ 0] Element | Value

BPSim PropertyParameter Values dialog fields

Field/Option/Button	Action
Property	This table lists the properties defined for the process, and shows the minimum and maximum possible values for each property for the whole process. If you click on the expansion box for a property, the table shows the minimum and maximum values for the property at each activity or event (element) during the process.
Token Number	Type the number of the 'token' to examine; this number must lie within the range shown to the right of the field. A 'token' is an independent trigger, such as a customer or an order entering a business and initiating the business process under review. There can be any number of customers or orders each having no relationship with any other customer or order, and each potentially following a

	different route through the business process. Where there is only one possible instance of one possible trigger event, such as an on/off switch being thrown, the token is regarded as being 0.
Query	Click on this button to initiate a query on the token simulation, to populate the 'Group by Element' and 'Group by Property' tabs.
Group by Element	Displays the results from the perspective of how a property parameter's value changes within a selected element. The tab shows a list of the elements in the process and, for each element, the value of each property applied in the element on each iteration of the simulation. Using the 'Toggle **Filter** Bar' option on the header bar, you can refine the display to show only a particular property and see how often it is used by the element and with what values.
Group by Property	Displays the results from the perspective of how the value of each property changes during the whole process. The tab shows a list of properties that are applied during the process and, for each property, the value in each activity (element) on each iteration of the process.

Examples

In the EAExample Model, you can study two examples of generating information on property parameters from a simulation of a BPMN business process model. These will demonstrate how you define property parameters in the configuration, based on the model. You can initially just run a simulation on each example, and examine the output as described here. You can then examine the business processes and configurations themselves, and change or add to the property parameters provided.

The examples are described in the *Tracking Property Values - Examples* topic. Briefly, they are:

- 'Fibonacci' - a very simple recursive business process that calculates a series of Fibonacci numbers through ten iterations; you can see how the property parameters increment in each iteration through the elements of the process (in Example Model > Model Simulation > BPSim Models > Fibonacci)
- 'Car Repair' - a more complex and realistic process that represents what might happen when a series of individual 'walk-in' customers bring vehicles into a car repair shop for estimation and repair (in Example Model > Model Simulation > BPSim Models > Car Repair Process)

There is also a small example of the behavior of the Time Parameters (in Example Model > Model Simulation > BPSim Models > Time Parameter).

Notes

- If a BPSim configuration contains result requests and a Custom simulation is performed on it, the 'BPMN Simulation Report' shows only the built-in parameters requested in the configuration; in contrast, the 'BPSim PropertyParameter Values' dialog lists all property parameters regardless of any result request settings or type of simulation

Tracking Property Parameter Values - Examples

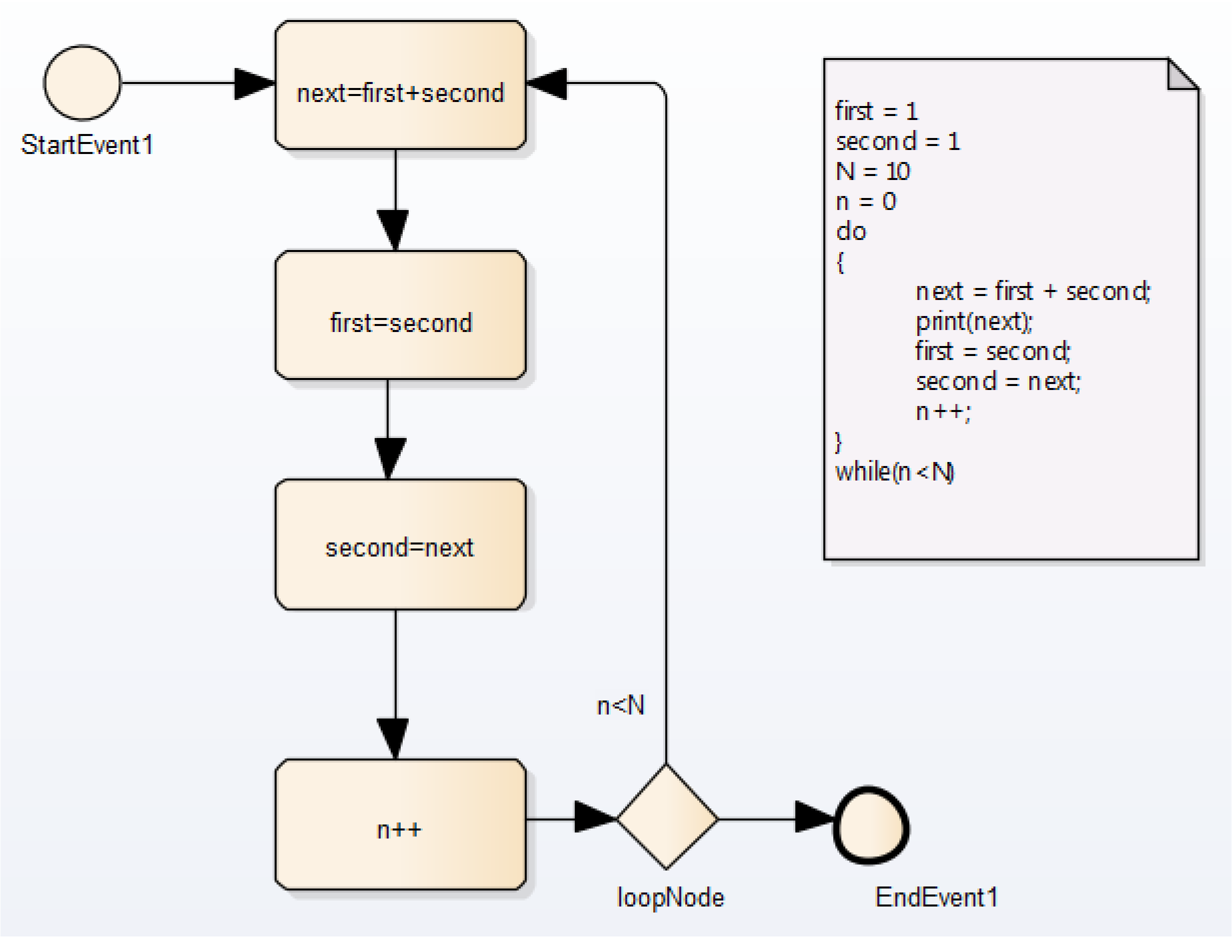
To help you learn about the facility for generating information on property parameters from a simulation of a BPMN business process model, Sparx Systems provide two examples that you can explore in the EAExample model. These are the:

- Fibonacci process - a very simple example to help you become familiar with the parameter tracking facilities
- Car Repair process - a more complex example that you can manipulate to see how a real-life process might be investigated

At the end of this topic is a section that briefly discusses how you might work with an integer-based process containing parameters that are initialized by 'real' distributions, and a section describing the example of the behavior of the Time Parameters.

The Fibonacci Example

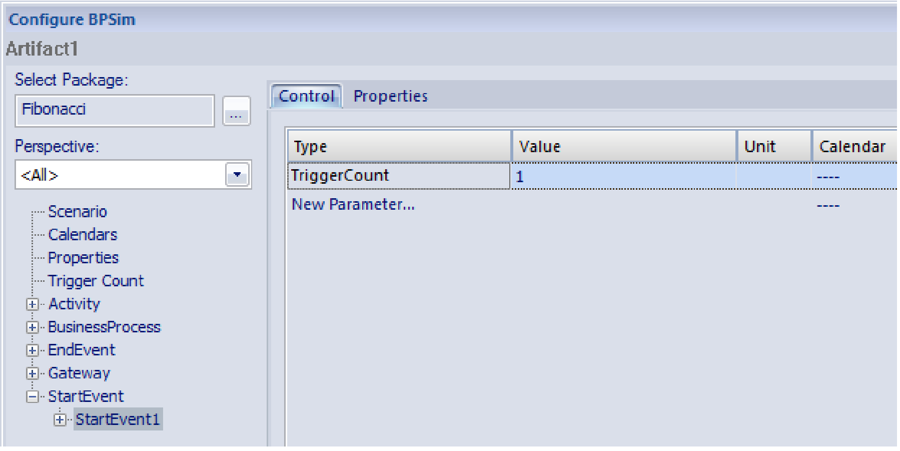
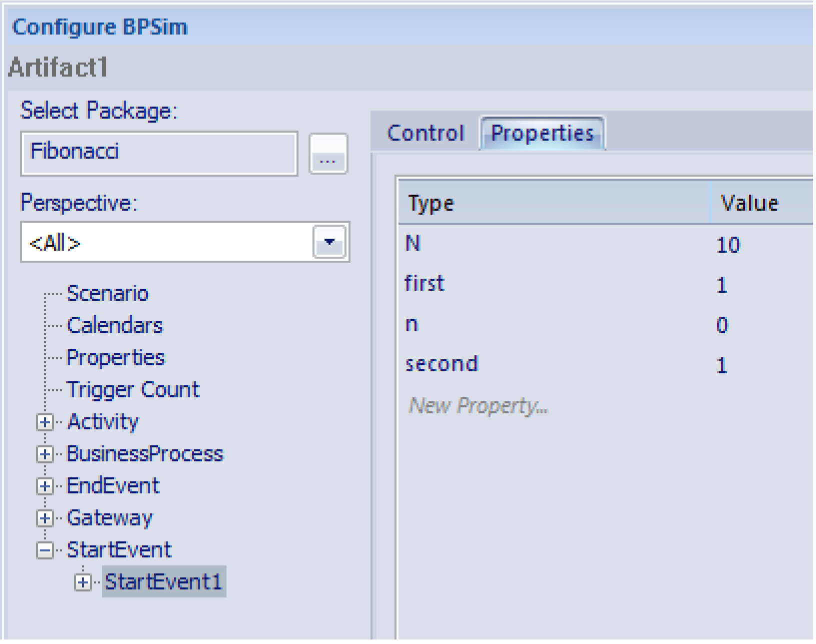
This is a very simple recursive business process that calculates a series of Fibonacci numbers through ten iterations; you can see how the property parameters increment in each iteration through the elements of the process. Open Example Model > Model Simulation > BPSim Models > Fibonacci.

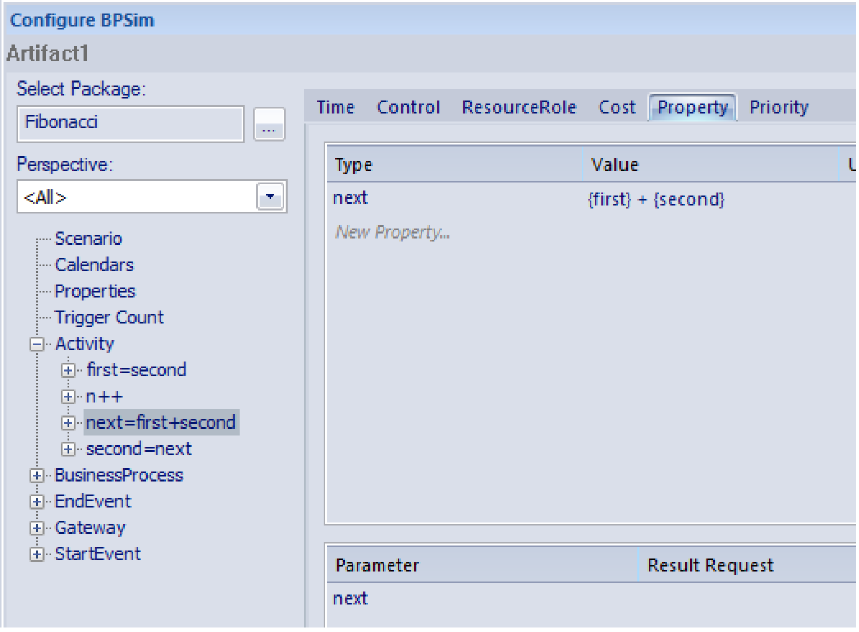
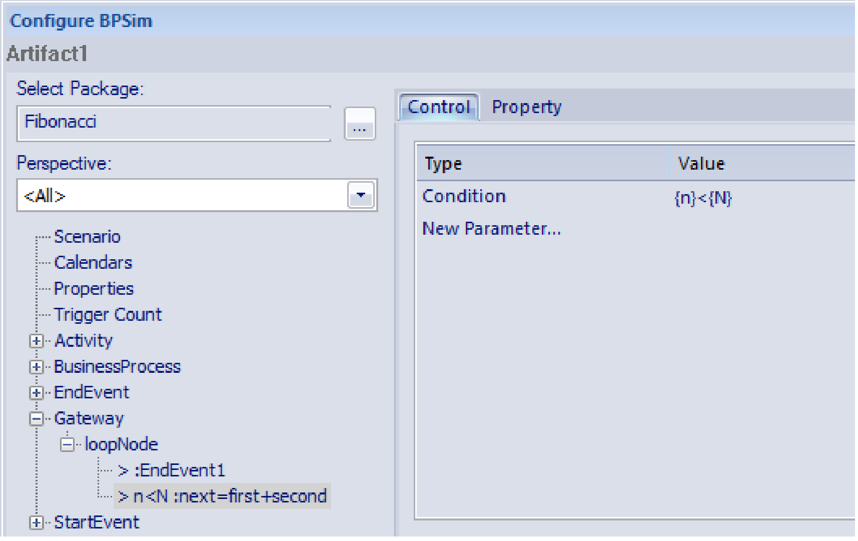


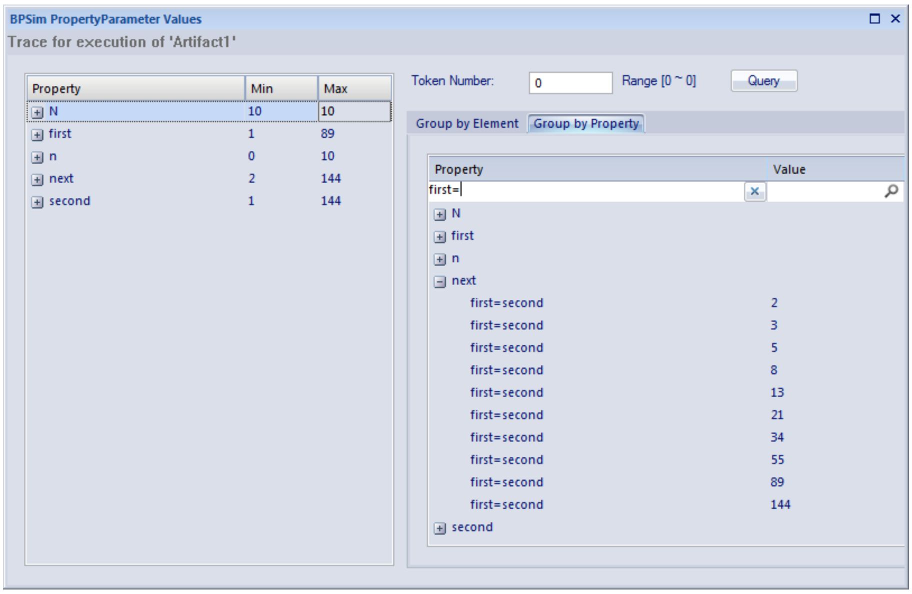
```
first = 1
second = 1
N = 10
n = 0
do
{
        next = first + second;
        print(next);
        first = second;
        second = next;
        n++;
}
while(n < N)
```

The pseudocode of the process is shown in the Notes element on the diagram. The statement 'print(next)' will output the number series 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

The BPSim configuration for this process is set up as described here.

Step	Action
1	 Configure BPSim Artifact1 Scenario Calendars Properties Trigger Count Activity BusinessProcess EndEvent Gateway StartEvent StartEvent1 Control Properties New Parameter... Select Package: Fibonacci Perspective: <All> Type Value Unit Calendar TriggerCount 1 ---- On the 'Control' tab for the StartEvent element, set 'TriggerCount' to '1', and on the 'Properties' tab create and initialize the properties: • 'N' as '10' • 'first' as '1' • 'second' as '1' • 'n' as '0'  Configure BPSim Artifact1 Scenario Calendars Properties Trigger Count Activity BusinessProcess EndEvent Gateway StartEvent StartEvent1 Control Properties New Property... Select Package: Fibonacci Perspective: <All> Type Value N 10 first 1 n 0 second 1
2	Now define the properties for each of the Activities in the process, on the 'Properties' tab. Note that the values for these properties are derived from **Expressions**, the components of which must be enclosed in braces - {xxx}. For the Activity: • next=first+second - set the property 'next' and define the value as the Expression {first} + {second}

	 Configure BPSim Artifact1 Time Control ResourceRole Cost Property Priority Scenario Calendars Properties Trigger Count Activity first=second n++ next=first+second second=next BusinessProcess EndEvent Gateway StartEvent Select Package: Fibonacci Perspective: <All> Type Value next {first} + {second} New Property... Parameter Result Request next • first=second - set the property 'first' and define the value as the Expression {second} • second=next - set the property 'second' and define the value as the Expression {next} • n++ - set the property 'n' and define the value as the Expression {n} + 1
3	Set the 'Condition' property parameters for the two Sequenceflow connectors issuing from the 'loopNode' Gateway element, on the 'Control' tab. Expand the Gateway | loopNode element and for the link to: • next=first+second - set the Control parameter to 'Condition' and define the value as the Expression {n}<{N}  Configure BPSim Artifact1 Control Property Scenario Calendars Properties Trigger Count Activity BusinessProcess EndEvent Gateway loopNode > :EndEvent1 > n<N :next=first+second StartEvent Select Package: Fibonacci Perspective: <All> Type Value Condition {n}<{N} New Parameter... • EndEvent1 - set the Control parameter to 'Condition' and define the value as the Expression {n}=>{N}
4	Having completed the configuration, click on the **Run button** on the 'Configure BPSim' dialog and on the 'BPSim Simulation Controller' dialog, selecting a Standard simulation. When the simulation is complete, click on the **Attributes button**.

	On the 'BPSim PropertyParameter Values' dialog, set the 'Token number' field to '0' and click on the **Query button**.
5	Now examine the values of the 'next' property on entering the first=second Activity in each iteration of the simulation. Click on the 'Group by Property' tab and expand the 'next' item. The list of values is long, so right-click on the column headers and select the 'Toggle **Filter** Bar' option. Under the 'Property' column heading, type 'first='. This filters the list to show only the property parameter values on entering the first=second Activity. 

BPSim PropertyParameter Values

Trace for execution of 'Artifact1'

Property	Min	Max
N	10	10
first	1	89
n	0	10
next	2	144
second	1	144

Token Number: 0 Range [0 ~ 0] Query

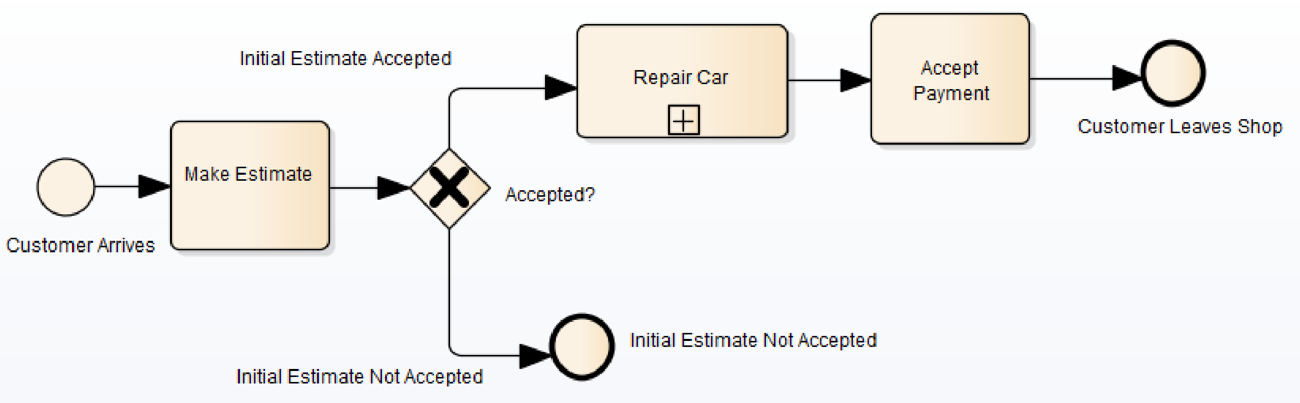
Group by Element | Group by Property

Property	Value
first=	
N	
first	
n	
next	
first=second	2
first=second	3
first=second	5
first=second	8
first=second	13
first=second	21
first=second	34
first=second	55
first=second	89
first=second	144
second	

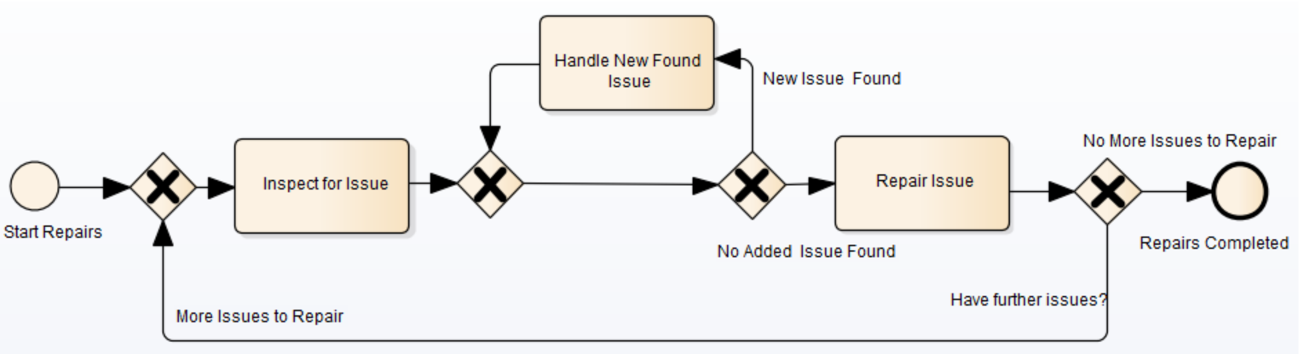
The Car Repair Example

This more complex example is based on a realistic model of a car repair process, where a number of individual customers request an estimate for repair and either proceed with the repair or decline to continue; you can see how the property parameters vary as different decisions are made during the process. Open Example Model > Model Simulation > BPSim Models > Car Repair Process.

The overall process is represented by this diagram:


Initial Estimate Accepted
Repair Car
Accept Payment
Customer Leaves Shop
Make Estimate
Accepted?
Customer Arrives
Initial Estimate Not Accepted
Initial Estimate Not Accepted

The *Repair Car* Activity is a composite element that contains this sub process diagram:



Step	Action
1	In the **Project Browser**, expand the 'BPSim' child Package underneath the 'Car Repair Process' Package, and double-click on the 'Scenario 1: Main Flow' Artifact. The **Configure BPSim window** displays. On the 'Scenario' tab look at the 'Duration' field; this has been set to 2 days and 12 hours (that is, 60 hours).
2	In the process hierarchy on the left of the window, expand the 'Start Event' category and click on 'Customer Arrives'. Select the 'Control' tab and look at the 'InterTriggerTimer' parameter, which has the value 24 minutes; that is, a customer arrives every 24 minutes (so over the 60 hour **duration**, 150 customers pass through the repair shop). Each customer enters the repair shop with one or more issues to be evaluated and repaired. The number of issues each customer presents can be randomly generated using one of the Distributions supported by BPSim. As the issue number is counted in discrete units (rather than measured on a continuous scale) we would use an 'integer' distribution. If you select the 'Properties' tab for the 'Customer Arrives' Start Event, you will see that the 'noOfIssues' property value is initialized from a Poisson distribution with a mean of 3.
3	Now expand the decision Gateway 'Accepted?' and its connectors, in the process hierarchy. 'Initial Estimate Accepted' has a Control parameter 'Probability', set to 0.67. The alternative connector, 'Initial Estimate Not Accepted' has a similar Control parameter 'Probability' set to 0.33. That is, we expect an average of one issue in three to be withdrawn - or not pursued - by the customer.
4	Further into the process, when an issue is being assessed on the vehicle, there is a possibility of another issue being discovered. In the list of Gateway elements, the last 'unnamed element' has two paths: 'New Issue Found' and 'No Added Issue Found'. Click on each of these and look at the 'Control' tab; the 'Probability' parameter for 'New Issue Found' is set to 0.25 and, for 'No Added Issue Found', to 0.75. So on average, for every four issues reported and assessed, one new one is discovered. The 'New Issue Found' path takes the process to the 'Handle New Found Issue' Activity, which adds 1 to the number of issues to be processed for the current customer. Expand the Activity group, and click on the 'Handle New Found Issue' element and on the 'Property' tab. You will see that the property 'noOfIssues' here has the Expression value {noOfIssues} **+ 1**.
5	When a problem with the vehicle is resolved, the 'Repair Issue' Activity deducts 1 from the number of issues to be repaired for the current customer. Click on the 'Repair Issue' element in the Activity group and on the 'Property' tab. You will see that the property 'noOfIssues' here has the Expression value {noOfIssues} **- 1**.
6	The value from the 'Repair Issue' Activity is tested at the 'Have Further Issues?' Gateway. Click on the 'More Issues to Repair' connector and on the 'Control' tab; the Condition parameter for following this path is set to the Expression value {noOfIssues} **> 0**; flow passes to the Gateway prior to

	the 'Inspect for Issue' Activity. Similarly, if you click on the 'No More Issues to Repair' connector and on the 'Control' tab, the Condition parameter for following that path is set to the Expression value {noOfIssues} **=< 0**, and flow passes to the 'Repairs Completed' End Event. Now that you have examined the process flow and configuration settings, you can run a simulation and review the results.
7	On the **Configure BPSim window**, click on the **Run button**, and then on the 'BPSim Simulation Controller' dialog click again on the Run button, selecting the 'Standard' simulation (however, the type of simulation makes no difference for reviewing property parameters). On the 'BPSim Simulation Controller' dialog, you can review the Token status (and see that an additional customer manages to enter the shop at the very last minute) but it is difficult to see exactly how this summary data resulted. Click on the **Attributes button** to obtain the detailed property parameter values information on the 'BPSim PropertyParameter Values' dialog.
8	On the left hand side of the dialog is a summary of the minimum and maximum values for the property parameter (attribute) for each element in the process. For example, for the 'Customer Arrives' element the 'noOfIssues' parameter has a minimum of 0 and a maximum of 8, as generated by the Poisson (3) distribution. In the 'Token Number' field, type in a number (N) between 0 and 150 to select for the Nth customer who entered the repair shop. Click on the **Query button** to obtain the property parameter values used in the process for that customer. Review the results on each of the two tabs: • On the 'Group by Element' tab, see how the attribute's value changes in each element; for example, for customer 24 the 'noOfIssues' parameter is initialized with a value of 4 by the random distribution, and the 'Inspect for Issue' Activity is called six times with the parameter value being adjusted to 3 for three of those calls before cycling to 1, and the 'Handle New Found Issue' Activity is called twice with the parameter value at 3 both times • On the 'Group by Property' tab see how the parameter value changes as the process cycles through the Activities to completion, starting at 4, being adjusted between 3 and 4 a number of times and then decrementing to 0 at the end
9	Continue to explore the results as you wish, selecting different customers (Tokens). You can also return to the BPSim configuration and change the parameter initializations and add new ones, or change the decision points, to experiment with the process.

Responding to real numbers in the simulation of an integer-based process

In some cases, you might need to generate property parameter values using a distribution that returns 'real' numbers when the activities in the process operate with integers or you want to see what impact forcing integer values has on the process.

One mechanism to apply in such cases is to set conditions to avoid absolute numbers. So, for example, you might have a counter that decrements by 1, that is initialized to a 'real' number. If you set a condition to 'value==0' (equals 0) or 'value !=0' (does not equal 0), the two conditions might never be **True** or might always be True, respectively, causing an infinite loop. To avoid that, in the conditions you would use operators such as:

'value > 0'

'value < 0'

'value >= 0'

'value <= 0'

Another mechanism is to edit the code template used by the BPSim engine, to intercept and replace the real numbers provided to specific parameters with integers, like this:

17. Select the 'Package | Code Engineering | Code Generation Templates' menu option.
18. In the **Code Template Editor**, in the 'Language' field, click on the drop-down arrow and select 'MDGBPSimExecutionEngineExtension'.
19. In the list of (Java) templates, click on 'MDGBPSimExecutionEngineExtension Compute Value'. The template contents display in the 'Template' panel.
20. Find this line:

    ```
    double %bpsimPropertyParameterName% = (double) distribution.next();
    ```

    Change it to:

    ```
    %if bpsimPropertyParameterName == "noOfIssues" or  bpsimPropertyParameterName == "noOfVisitors"%
    double %bpsimPropertyParameterName% = (int) distribution.next();
    //double %bpsimPropertyParameterName% = Math.ceil( distribution.next() );
    //double %bpsimPropertyParameterName% = Math.floor( distribution.next() );
    //double %bpsimPropertyParameterName% = Math.round( distribution.next() );
    %else%
    double %bpsimPropertyParameterName% = (double) distribution.next();
    %endIf%
    ```

21. Replace the property parameter names with your own property parameters.
22. Click on the **Save button**, close the Code Template Editor and reload the project.

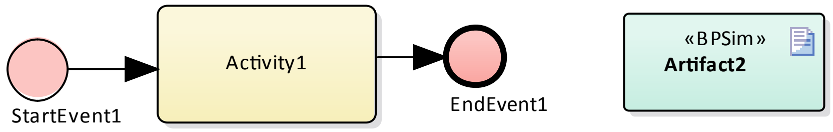
As presented, for each specified parameter the code template will simply replace any 'real' number initialized by the distribution with an integer. If you prefer, you can use one of the commented lines instead:

- Math.ceil() will take the 'real' number and convert it to the next highest integer
- Math.floor() will take the 'real' number and convert it to the next lowest integer
- Math.round() will take the 'real' number and round it up or down depending on whether it is greater than or less than n.5

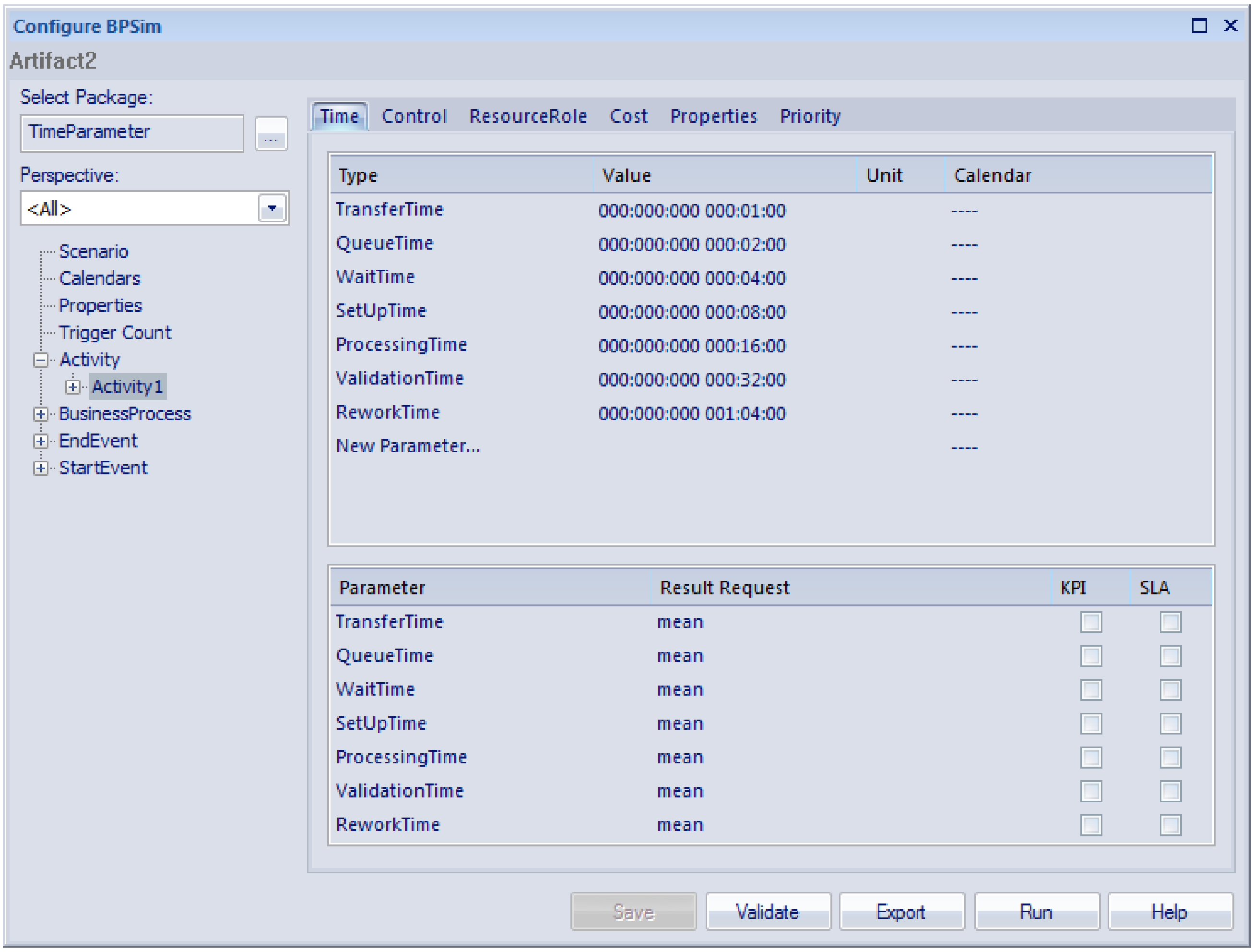
Time Parameter Behavior

In the BPSim configuration, you can set a number of Time parameters for an Activity, such as Queue Time and Wait Time. You can also set a Result Request on each of these, for a Custom simulation. However, the BPSim simulation engine combines these parameters into a single 'Processing Time' quantity.

Consider the simple model TimeParameter in the Example Model (Example Model > Model Simulation > BPSim Models > Time Parameter), represented by this diagram:



If you double-click on the Artifact2 element, the 'Configure BPSim' dialog displays. Click on the Activity1 element in the diagram to expand the Activity group, to select Activity1 in the hierarchy at the left of the dialog, and to display the first tab, 'Time', for the element in the configuration, as shown.



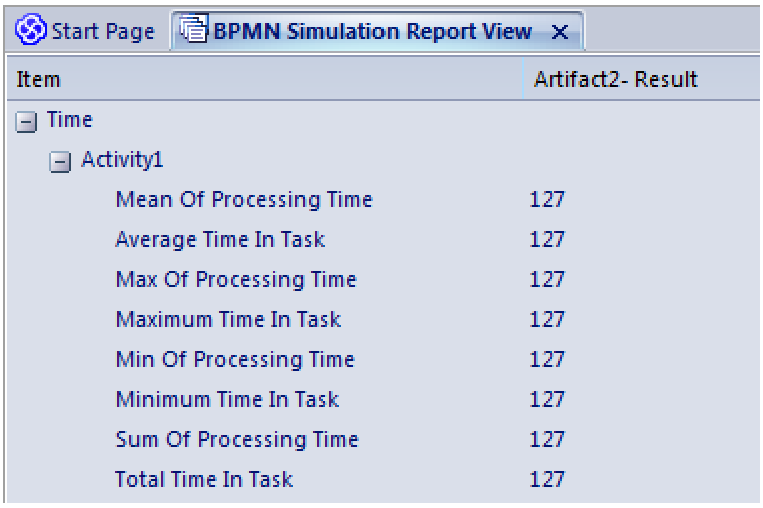
Select Package: TimeParameter

Type	Value	Unit	Calendar
TransferTime	000:000:000 000:01:00		----
QueueTime	000:000:000 000:02:00		----
WaitTime	000:000:000 000:04:00		----
SetUpTime	000:000:000 000:08:00		----
ProcessingTime	000:000:000 000:16:00		----
ValidationTime	000:000:000 000:32:00		----
ReworkTime	000:000:000 001:04:00		----
New Parameter...			----

Parameter	Result Request	KPI	SLA
TransferTime	mean	□	□
QueueTime	mean	□	□
WaitTime	mean	□	□
SetUpTime	mean	□	□
ProcessingTime	mean	□	□
ValidationTime	mean	□	□
ReworkTime	mean	□	□

Note that in the upper panel there are seven system-supplied 'Time' parameters, which have been given initial values of - in order - 1, 2, 4, 8, 16, 32 and 64 minutes (the 64 minutes is 1 hour and 4 minutes). Note also that in the lower panel, each of these has a Result Request for the mean runtime value of the parameter.

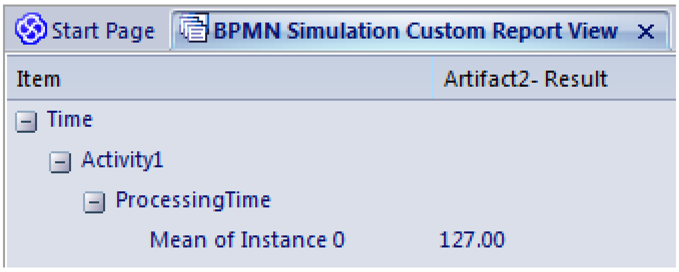
Click on the **Run button**, and on the 'BPSim Simulation Controller' dialog click on the Run button and select 'Standard Simulation'. The simulation is configured to cycle through the process once. When the simulation is complete, click on the **Open Result button**, and on the 'BPMN Simulation **Report View**' right-click and select the 'Show Only Non-Empty Items' option. This gives you, for the Activity1 element on which the parameters were set, these results:



Item	Artifact2- Result
Time	
Activity1	
Mean Of Processing Time	127
Average Time In Task	127
Max Of Processing Time	127
Maximum Time In Task	127
Min Of Processing Time	127
Minimum Time In Task	127
Sum Of Processing Time	127
Total Time In Task	127

All of these derived results are 127 minutes, the sum of the initial values of the original seven 'Time' parameters. The individual parameters are not processed separately.

If you return to the 'BPSim Simulation Controller' dialog and click on the Run button, selecting 'Custom Simulation' this time, the Open Result button displays the 'BPMN Simulation Custom Report View'. In the configuration, the Result Requests were for the mean values of the seven parameters. In the Report View for the simulation, you only see the mean of the single aggregated parameter, ProcessingTime, as 127 minutes.



Item	Artifact2- Result
⊟ Time	
⊟ Activity1	
⊟ ProcessingTime	
Mean of Instance 0	127.00

Compare BPSim Configurations

When you develop a BPSim configuration, you can define a wide range of parameters to set prior to running simulations and observing the effects of those settings, and of changes in selected settings. To facilitate using and managing multiple 'what-if' scenarios, it is recommended that you create copies of the original configuration (as Artifact elements) and make the setting changes in the copies.

A useful facility in creating variations of a configuration is to apply inheritance, whereby the data and parameters you do NOT intend to vary are held in one configuration, and only those parameters that you change are held in another. The 'variable' configuration uses (inherits) the common data held in the base configuration, so you do not have to re-create that common data in the 'variable' configuration.

You can then run simulations on the changed configurations and on the original 'baseline' and compare the simulation reports to see what differences occurred in the run-time variables, and then run and display the comparisons of the configurations to see what changes in parameter settings gave rise to those run-time differences.

By running simulations under the original and copy configurations, comparing the results and the changes that caused the results, and modifying the model accordingly, you can achieve a very high degree of control in streamlining the business process you are developing.

Access

Context Menu	On diagram or **Project Browser** | Right-click BPSim Artifact element | Show BPSim Configuration

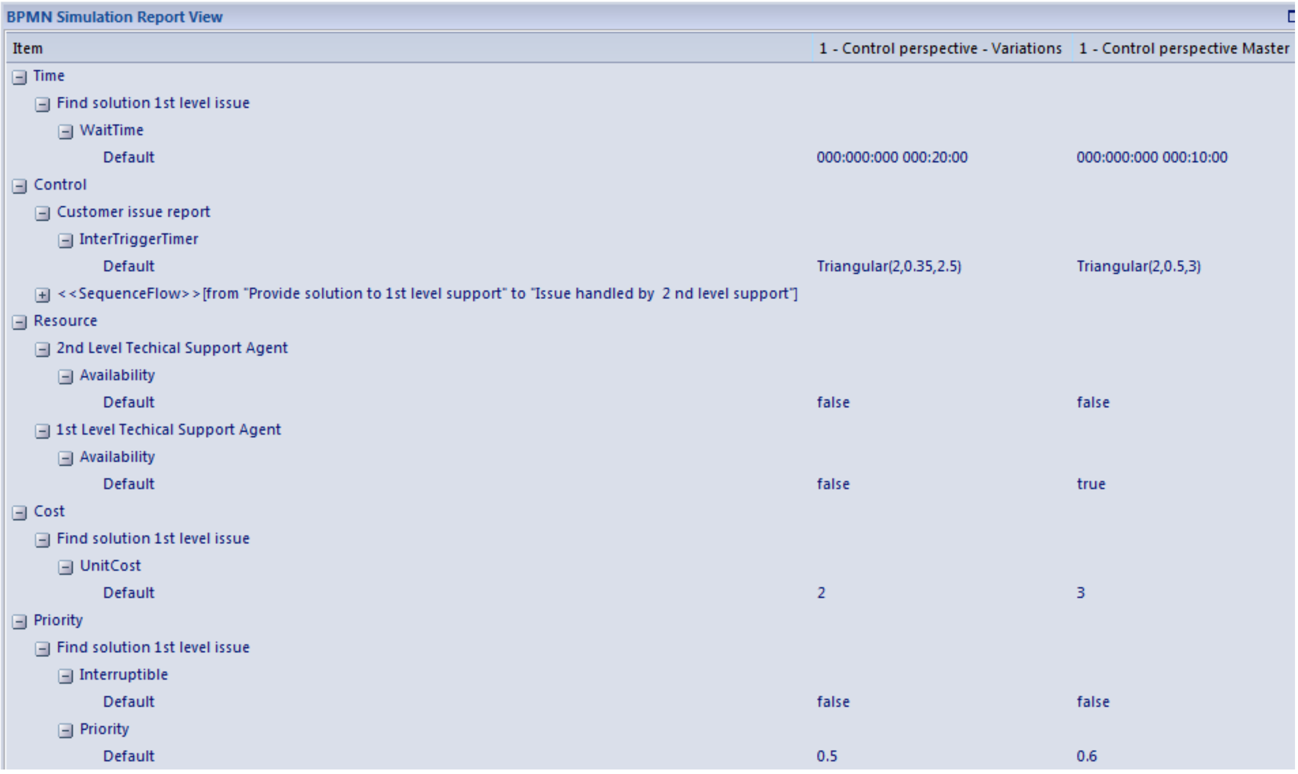
The BPMN Simulation Configuration Report View

This view initially displays a hierarchy of the parameters in the selected configuration that have values and, in the column under the configuration name, the values set for those parameters. You can expand or collapse individual tab and parameter items using the +/- expansion/contraction boxes, or the hierarchy as a whole by right-clicking on the view and selecting the 'Collapse All' or 'Expand All' options. To review the results more easily, you can also drag the tab off the workspace to become a floating display, and expand the display to full screen size.

To compare two (or more) configurations, click on another configuration Artifact element in the **Project Browser** or diagram and either:

- Drag it onto the report view or
- Select the 'Show BPSim Configuration Results' option

The parameter hierarchy now contains any additional parameters in that configuration, and its parameter values display in a column to the left of the original configuration values.

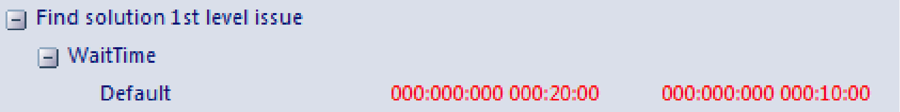


BPMN Simulation Report View

Item	1 - Control perspective - Variations	1 - Control perspective Master
Time		
Find solution 1st level issue		
WaitTime		
Default	000:000:000 000:20:00	000:000:000 000:10:00
Control		
Customer issue report		
InterTriggerTimer		
Default	Triangular(2,0.35,2.5)	Triangular(2,0.5,3)
<<SequenceFlow>>[from "Provide solution to 1st level support" to "Issue handled by 2 nd level support"]		
Resource		
2nd Level Techical Support Agent		
Availability		
Default	false	false
1st Level Techical Support Agent		
Availability		
Default	false	true
Cost		
Find solution 1st level issue		
UnitCost		
Default	2	3
Priority		
Find solution 1st level issue		
Interruptible		
Default	false	false
Priority		
Default	0.5	0.6

You can review and manipulate the information on the report using options available from a right-click context menu.

BPMN Simulation Configuration Report Options

Option	Description
Collapse All	Select this option to collapse the parameter hierarchy down to just the parent tab names.
Expand All	Select this option to expand the parameter hierarchy to the lowest value type.
Show Only Different items	(When you have two or more configurations shown.) Select this option to restrict the display to those parameters where the values differ between the configurations. Click on the option again to deselect it.
Highlight Different Items	(When you have two or more configurations shown and where their parameter values are different.) Displays the differing parameter values in red. This option is disabled if you select the 'Show Only Different Items' option.  Find solution 1st level issue WaitTime Default 000:000:000 000:20:00 000:000:000 000:10:00
Show Only Non Empty Items	Select this option to filter the display to show only parameters that have a specific value other than 0.
Remove Scenario	(When you have selected an item within a configuration.) Select this option to remove the whole configuration from the report.

Edit Parameter	(When you have selected an item within a configuration.) Select this option to display the appropriate tab in the **Configure BPSim screen**, with the parameter and value highlighted, and edit the value. When you have edited the value and clicked off it, the **Save button** is enabled; click on this button to return to the **Report View**. The hierarchy is collapsed, but when you expand it again the selected parameter value shows the change you made.

BPSim Charts

The Artifacts page of the **Diagram Toolbox** provides two icons specifically to generate charts that reflect selected results from BPSim simulations. These are:

- BPSim Result Chart - to generate a Chart that reflects selected results from a series of standard BPSim simulations
- BPSim Custom Result Chart - to generate a Chart that reflects results from a series of customized BPSim simulations

As for other Chart Artifacts, both BPSim Chart types can be quickly configured to display the simulation results in variations of a line chart, two-dimensional bar chart or 3-dimensional bar chart.

Prerequisites

To populate the Charts created from the BPSim Artifacts, you select the Result Artifacts created during the simulation of each configuration that you want to show. Therefore, the initial simulations must be performed first, and the Report Artifacts generated.

Access

Display the 'Artifacts' page of the diagram toolbox using any of the methods outlined in the table below.

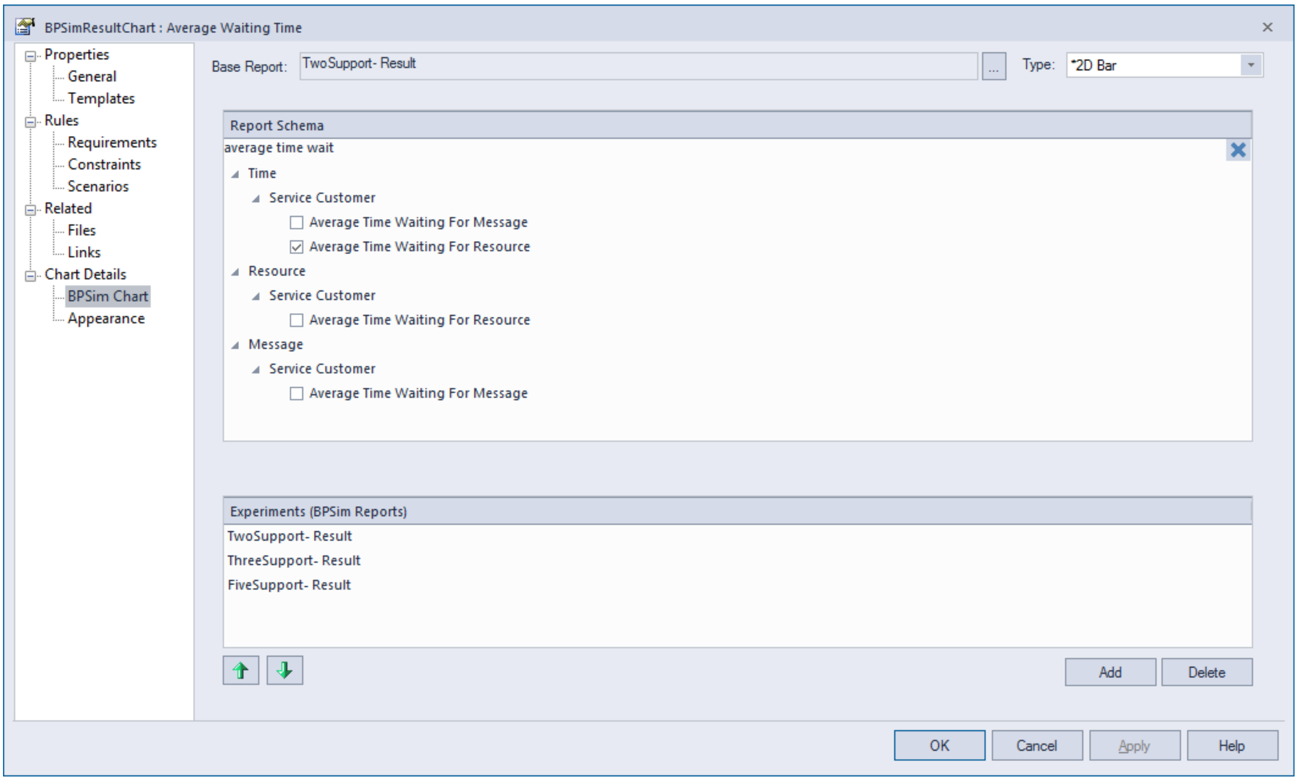
Then, drag the BPSim <type> Chart Artifact icon onto the diagram - a new chart element is created.

Double-click on the new chart element to open the 'Properties' dialog, showing the 'BPSim Chart' page.

Ribbon	Design > Diagram > Toolbox > Artifacts
Menu	Diagram | Toolbox > Artifacts
Keyboard Shortcuts	**Alt** + 5 | Aritfacts
Other	You can display or hide the **Diagram Toolbox** by clicking on the » or « icons at the left-hand end of the Caption Bar at the top of the **Diagram View**.

Select Results to Display In Chart

Complete the fields on the 'BPSim Chart' page of the Chart 'Properties' dialog.



BPSimResultChart : Average Waiting Time

Base Report: TwoSupport- Result

Type: *2D Bar

Report Schema

average time wait

- Time
  - Service Customer
    - □ Average Time Waiting For Message
    - ☑ Average Time Waiting For Resource
- Resource
  - Service Customer
    - □ Average Time Waiting For Resource
- Message
  - Service Customer
    - □ Average Time Waiting For Message

Experiments (BPSim Reports)

TwoSupport- Result
ThreeSupport- Result
FiveSupport- Result

Field/Option	Action
Base Report	Click on the ... button to display the 'Select <<BPSimReport>>Artifact' dialog, and select the report Artifact for the simulation results against which other simulation results will be compared. This Results Artifact is also added to the 'Experiments' panel as the first in the list of report results to be compared.
Type	Click on the drop-down arrow and select the format type of the Chart in which to display the results - 2D Bar, 3D Bar or Line. After you have specified the report parameters to compare, you can select the 'Appearances' page of the dialog and define the appearance of the bar chart or Time line graph.
Report Schema	Expand the hierarchy as necessary and select the checkbox against each property to display on the chart. Each property will be represented by a separate line or group of bars on the chart. Usually you would select similar objects (such as different resources) and the same single property for each object (such as the degree of utilization of the resource). You have a wealth of properties to examine and compare, but any more than a couple on the same chart makes the chart hard to read.
Experiments (BPSim Reports)	This panel lists the simulation report results (as BPSim Result Artifacts) that you have selected to compare using the chart, in the sequence in which their selected parameter will be shown on the Chart. Usually, the Base Report remains as first in the list and the result for its parameter is shown at the left of the Chart. If you want to change the sequence, click on the Result Artifact name and click on the appropriate **Up/Down green arrow button**. To add further Result Artifact names to the list, click on the **Add button** and browse for and select the Artifact from the 'Select <<BPSimReport>> Artifact' dialog.

A BPSim Chart Example

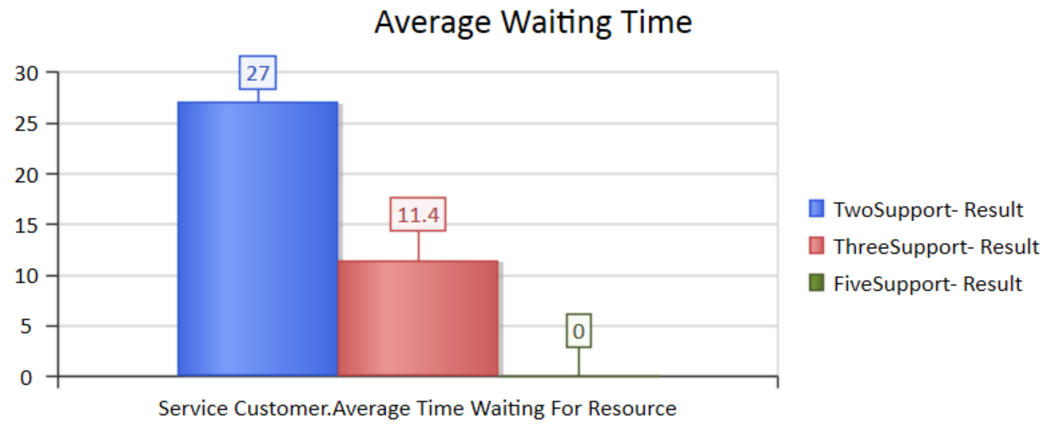
In the Help Desk Phone Support example, we created three BPSim artifacts to do what-if analysis: How many supports do we need to answer the customer's questions over the phones in a budget way.

We started with 2 supports, then tried 3 and 5 supports. After simulation, we've got BPSim Report based on different configurations: TwoSupport- Result, ThreeSupport- Result, FiveSupport-Result.

Here is the steps to create a chart to compare the customer's average time waiting for a support:

- Create a BPSim Result Chart on the diagram, name it *Average Waiting Time*
- Double click it to open the properties window | Open the BPSim Chart tab
- Click "..." button to select a Base Report, from which we define the schema (legends) we want to use in the chart. Select *TwoSupport-Result*
- Choose the following schema:

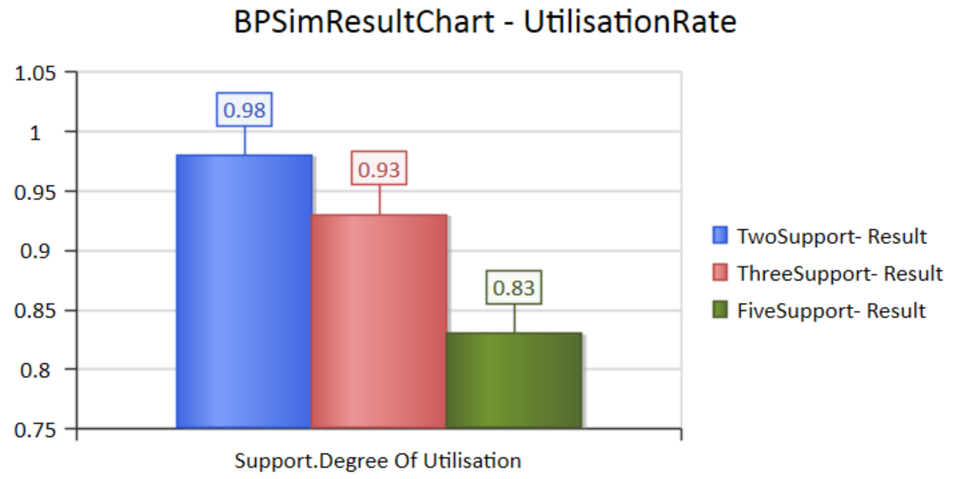
  - Time | Service Customer | Average Time Waiting For Resource
- Click "Add" to add another two BPSim Reports: *ThreeSupport- Result* and *FiveSupport- Result*
- OK, adjust the size of the chart element. This chart gave us very straight information.



Here is the steps to create a chart to compare the degree of utilisation of the supports

- Create a BPSim Result Chart on the diagram, name it *Utilisation Rate*
- Double click it to open the properties window | Open the BPSim Chart tab
- Click "..." button to select a Base Report, from which we define the schema (legends) we want to use in the chart. Select *TwoSupport-Result*
- Choose the following schema:

  - Resource | Support | Degree Of Utilisation
- Click "Add" to add another two BPSim Reports: *ThreeSupport- Result* and *FiveSupport- Result*
- OK, adjust the size of the chart element. This chart gave us very straight information.


BPSimResultChart - UtilisationRate
1.05
1
0.95
0.9
0.85
0.8
0.75
0.98
0.93
0.83
Support.Degree Of Utilisation
TwoSupport- Result
ThreeSupport- Result
FiveSupport- Result

BPSim Examples

This document contains many examples covering the BPMN modeling, BPSim Configuration and Analysis on the simulation results.

These examples can be accessed from EA Example model.

EABPSim Execution Engine is required to run the simulation.

Calendar Based Help Desk Phone Support Simulation

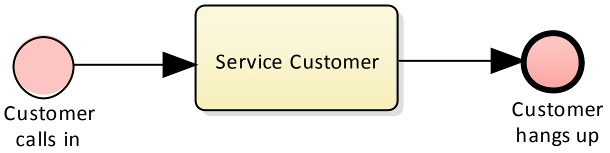
In this example, we create a very simple model to simulate the help desk phone support process based on calendar settings.

- We assume customer calls in different intervals in weekdays and weekends.
- We assume it took different processing times in weekdays and weekends.
- We assume we have different number of supports in weekdays and weekends.

We step-by-step model this process, then create calendars and configure the BPSim, which is simple enough so it is capable to be calculated with pen & paper. After that, we run the simulation to compare the result with the pen & paper analysis.

Create BPMN Model

The model itself is very simple, a Start Event, a Task and an End Event.



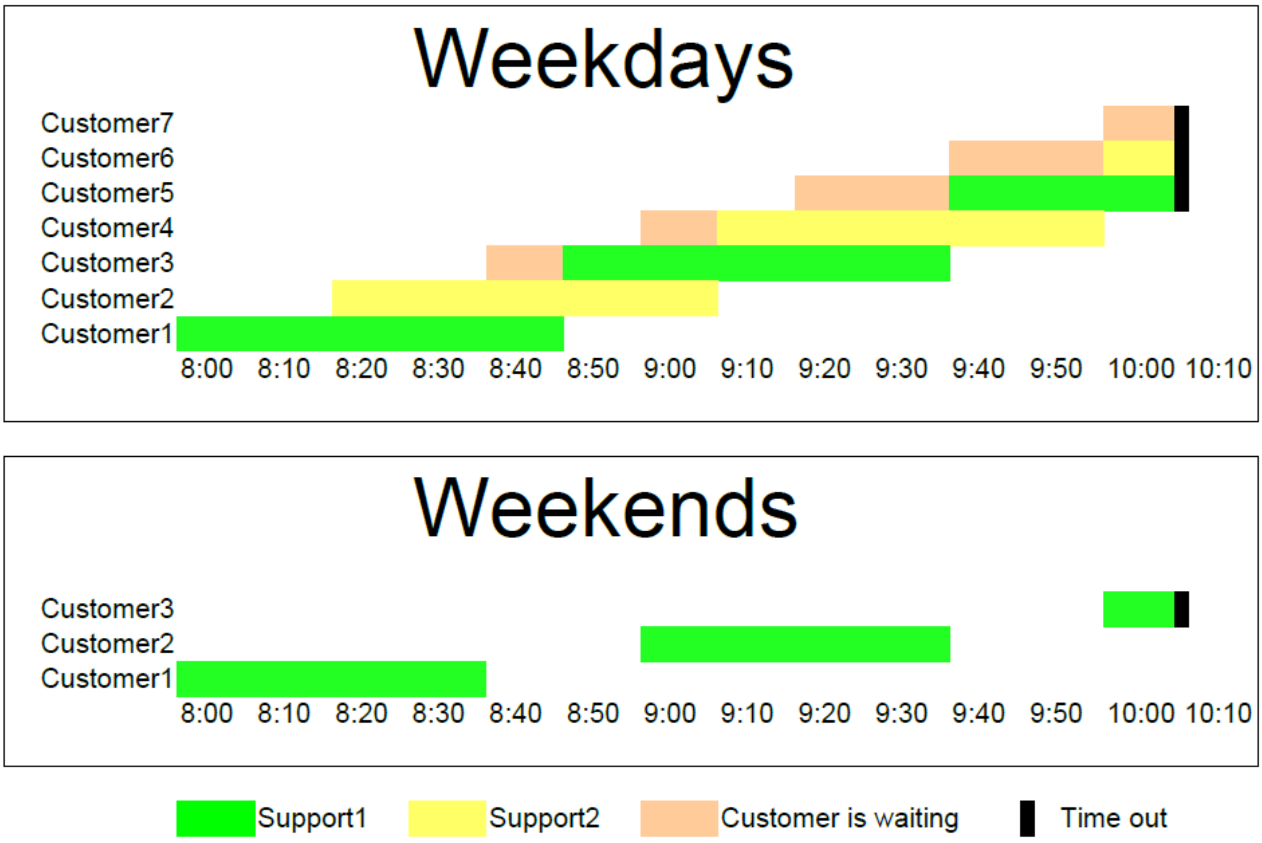
- Create a Start Event *Customer calls in*, sequence flow to
- an abstract task *Service Customer*, sequence flow to
- an End Event *Customer hangs up*

Create a BPMN2.0 Resource named *Support;* Create a BPMN2.0::ResourceRole inside *Service Customer*, give it name *support* and set the tag *resourceRef* to the Resource element *Support*.

Pen & Paper Analysis

We will use pen & paper to analyze the following case:

- Simulation **duration** 2 hours and 10 minutes from 8:00AM to 10:10AM
- Customer calls in every 20 minutes during weekdays
- Customer calls in every 60 minutes during weekends
- Service Customer takes 50 minutes during weekdays
- Service Customer takes 40 minutes during weekends
- There are 2 supports during weekdays
- There is 1 support during weekends



Seen from this result, it is already very complicated computation for such a simple model when resource constraints are applied.

On weekdays,

- 7 customers called every 20 minutes in a duration of 2 hours and 10 minutes
- 4 customer calls hung up normally
- 2 customer calls were interrupted because of timeout
- 1 customer call was not answered
- Support1 worked 130 minutes; Support2 worked 110 minutes

On weekends,

- 3 customers called every 60 minutes in a duration of 2 hours and 10 minutes
- 2 customer calls hung up normally
- 1 customer calls were interrupted because of timeout
- Support1 worked 90 minutes;

Now we will see how EABPSim can help.

BPSim Configuration

In this section, we first create the Calendars, then setup the Duration and Start parameters.

For element parameters, you can setup specify one or more calendars for a given parameter.

However, **if any calendar is set for a parameter value, a default value (without any calendar specified) must exist**. otherwise the simulation won't work.

The validation button on the Toolbar will automatically check this constraint for you.

Create BPSim Artifact and Set Package

- Open BPSIm Manager
- Create Artifact named ***Calendar** Based Support Process Simulation*
- Select Package of the model created

Then all the BPMN elements will be loaded in to the manager.

Calendars	Select *Calendars* in the tree to open *Calendars* tab, click **New button** to create a calendar, set the fields as follows and save with a name. • *Event Time* - *Start*: 08:00AM; Event Time - *End*: 05:00PM; Select *Weekly* and check Monday ~ Friday; *Start* Date choose to 4/1/2016; No end date; Save as "Weekdays" • *Event Time* - *Start*: 08:00AM; Event Time - *End*: 05:00PM; Select *Weekly* and check saturday and Sunday; *Start* Date choose to 2/1/2016; No end date; Save as "Weekends"
Start & Duration	**Duration** input 0000 002:10:00, which means 0 days, 2 hours and 10 minutes **Start** We will run 2 simulations with this artifact, one for Weekdays, one for Weekends. The only difference between the two simulations is the value for the Start parameter. • For Weekday simulation: input 1/07/2016 8:00AM, which means 1st of July, 2016, which is a Friday. • For Weekend simulation: input 2/07/2016 8:00AM, which means 2nd of July, 2016, which is a Saturday.
Customer arrivals	Select StartEvent | *Customer calls in,* activate *Control* tab, create following parameter values: • Type: InterTriggerTimer; Value: 100:000:000 000:00:00; **Calendar**: ---- (this default value is necessary) • Type: InterTriggerTimer; Value: 000:000:000 000:20:00; Calendar: Weekdays • Type: InterTriggerTimer; Value: 000:000:000 001:00:00; Calendar: Weekends
Processing Times	Select Activity| *Service Customer,* activate *Time* tab, create following parameter values:

	• Type: ProcessingTime; Value: 100:000:000 000:00:00; **Calendar**: ---- (this default value is necessary) • Type: ProcessingTime; Value: 000:000:000 000:50:00; Calendar: Weekdays • Type: ProcessingTime; Value: 000:000:000 000:40:00; Calendar: Weekends
Resources	Select Resource| *Support,* activate *Resource* tab, create following parameter values: • Type: Quantity; Value: 0; **Calendar**: ---- (this default value is necessary) • Type: Quantity; Value: 2; Calendar: Weekdays • Type: Quantity; Value: 1; Calendar: Weekends
Resource Selection (allocation)	Select Activity | *Service Customer*, activate *Resource Role* tab Ensure the bottom group shows **bpsim::getResource('Support', 1)** as expression. This expression is loaded from your BPMN model by default. You can do some advanced configurations for resource selection for a task.

Run Simulation

Run Simulation - Weekdays:

- Change the Start to 1/7/2016
- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation

A file with name ***Calendar** Based Support Process Simulation- Result* is generated;

This report file contains the result for a weekday simulation.

Run Simulation - Weekends

- Change the Start to 2/7/2016
- From the Toolbar, click the Run button to open the BPSim Controller dialog
- Run button | Standard Simulation

A file with name *Calendar Based Support Process Simulation- Result* is updated.

This report file contains the result for a weekday simulation.

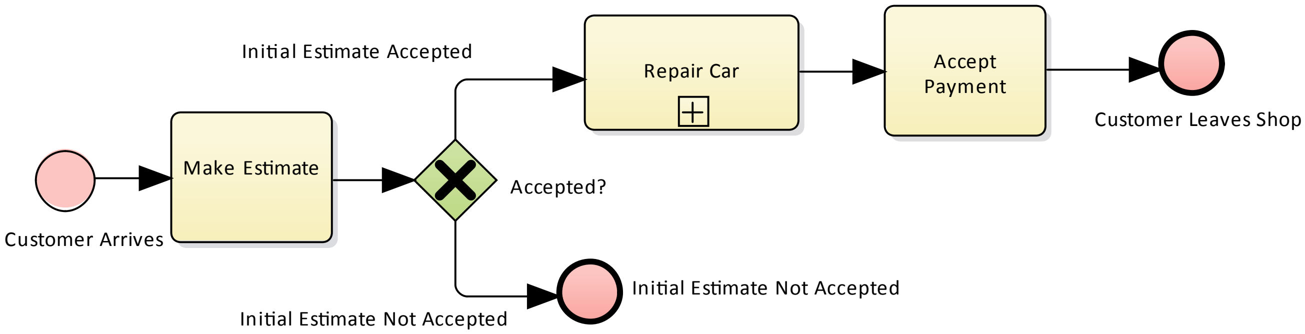
Open the Result file and the it match our analysis with pen & paper.

Car Repair Process

This example simulate the process of Car Repair shop. The BPSim configuration has following features:

- use a property parameter initialized by distribution to generate random number of issues for each customer
- use probability for acceptance of the initial estimate or not
- use probability for whether new issue will be found during repair
- increment/decrement the property parameter's value in each task
- use the property parameter's value on conditions of sequences outgoing from gateways
- simulate the customer arrivals in a given start and **duration**

Create BPMN Model

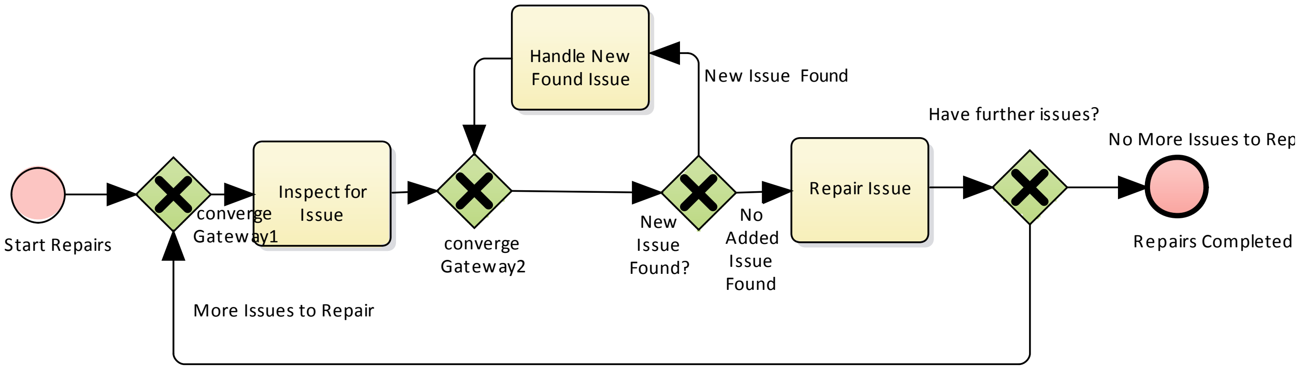

Initial Estimate Accepted
Repair Car
Accept Payment
Customer Leaves Shop
Make Estimate
Accepted?
Customer Arrives
Initial Estimate Not Accepted
Initial Estimate Not Accepted

Create the main process

- Create a Start Event *Customer Arrives*, sequence flow to
- an abstract task *Make Estimate*, sequence flow to
- an exclusive gateway *Accepted?*, sequence flows to

  - an end event *Initial Estimate Not Accepted*

  - a subProcess *Repair Car*
- from *Repair Car*, sequence flow to
- an abstract task *Accept Payment*, sequence flow to
- an end event *Customer Leaves Shop*



Create the sub process Repair Car

- Create a Start Event *Start Repairs*, sequence flow to
- an exclusive gateway *converge Gateway1*, sequence flow to
- an abstract task *Inspect for Issue*, sequence flow to
- an exclusive gateway *converge Gateway2*, sequence flow to
- an exclusive gateway *New Issue Found?*, sequence flow to
  - an abstract task *Handle New Found Issue*, sequence back to *converge Gateway2*
  - an abstract task *Repair Issue*, sequence flow to an exclusive gateway *Have further issues?*
- from gateway *Have further issues*, sequence flows to
  - an end event Repairs Completed
  - converge Gateway1

Configure BPSim

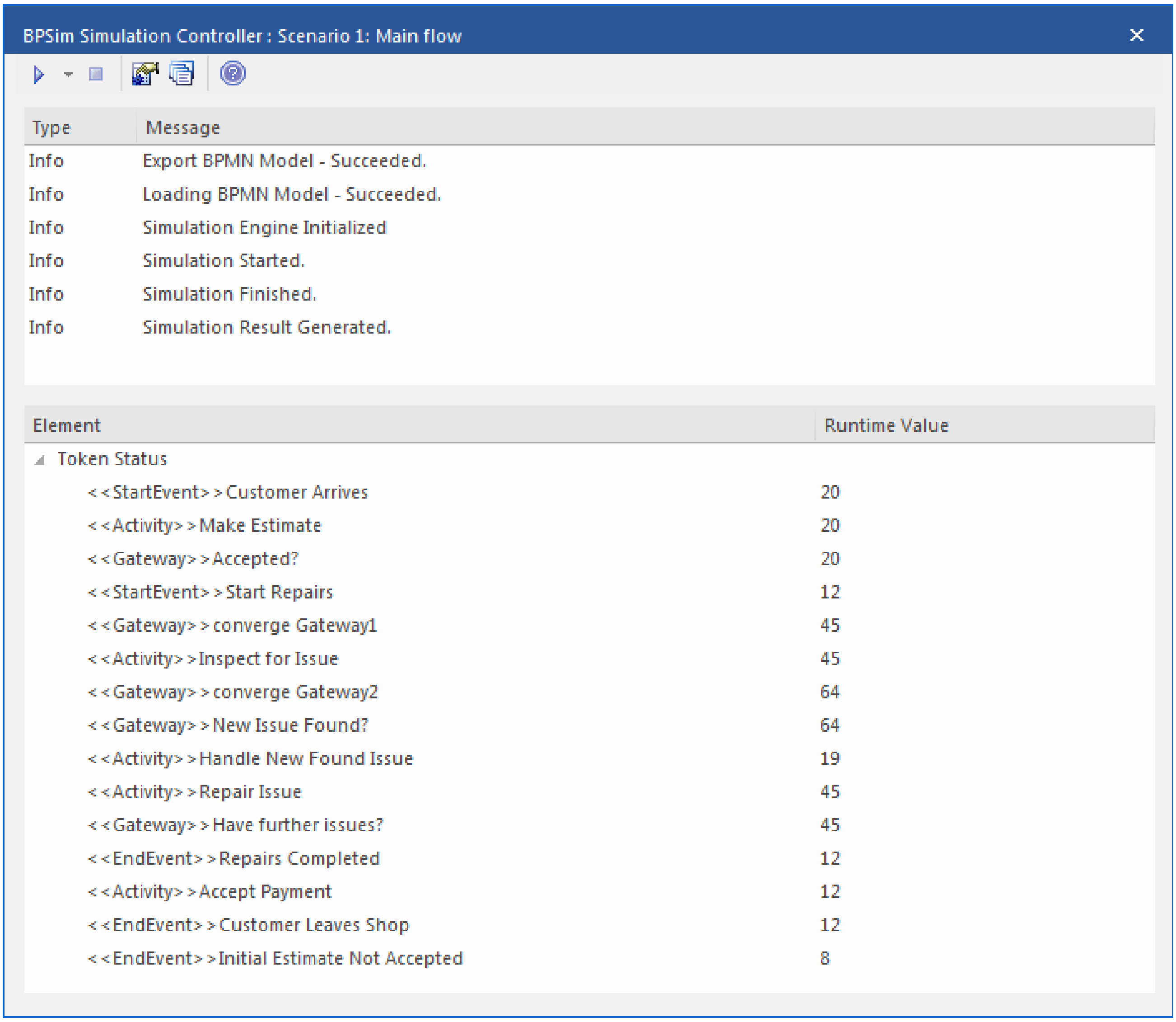
Artifact & Package	• Open BPSIm Manager • Create Artifact named "IntermediateEvent - Boundary - Error" • Select Package of the model created Then all the BPMN elements will be loaded in to the manager.
Start & Duration	We would like to simulate the processes in a car repair shop whose opening hour is from 9:00 am to 5 pm, which is 8 hours. We also suppose the customer walked in after 4:50 pm will not be served on that day. So the simulation's Start time is 9:00 am, **duration** is 7 hours and 50 minutes. Open Configuration Manager, click Scenario from the tree, then the Scenario page will be activated: • Start: choose any date and change the time section to be 9:00 AM • Duration: 0000 007:50:00 (Tips: the format is: dddd hh:mm:ss)
Customer Arrives	We would like to simulate the customer arrives in every 24 minutes. The first customer arrives at 9:00 AM, the last arrives at 4:36 PM (the customer arrives at 5:00 PM will not be served today because it is constraint by Duration). With a pen and paper, we can expect there are 20 customer served (9:00 AM to 4:36PM is 456 minutes, number of customers: 456/24 + 1 = 19 + 1 = 20). We will verify this with the simulation result later. Select *Customer Arrives*, click Control tab, add InterTriggerTimer with value: 24 minutes
Property Parameters	We suppose each customer's car may initially have different number of issues. This could be accomplished by random number generator. EABPSim provided many distributions to suit your needs. In this example, we use TruncatedNormal distribution to initialize the property noOfIssues. Tasks *Repair Issue* and *Handle New Found Issue* will decrement and increment the value of noOfIssues respectively.

	- Select *Customer Arrives*, click Properties tab, add noOfIssues, click "..." to open the parameter dialog, choose Distribution tab | select TruncatedNormal| Input 2, 1, 1, 1000 in the fields mean, standardDeviation, min, max respectively. **Important Note:** Some distribution like TruncatedNormal in this example return a floating point value, but the property is used as an integer. Setting the property's type is important especially in condition expressions when testing with equality. For example, the condition expression getProperty('noOfIssues') = 0 will almost never be satisfied because noOfIssues was initialized by a floating point distribution. ***Tips: Here is how you customize the type for property*** *After create the property and set a value, click "Save" button on the toolbar, then the created property name will be shown in the bottom list of Property tab, change its type from the default double to int.* - Select *Repair Issue* | click Properties tab, add noOfIssues, click "..." to open the parameter dialog, choose Expression tab | add "**{noOfIssues} -1**" to the Expression field - Select *Handle New Found Issue* | click Properties tab, add noOfIssues, click "..." to open the parameter dialog, choose Expression tab | add "**{noOfIssues} +1**" to the Expression field
Probability on Sequence flows	We assume 1/3 of the customers will not accept the initial estimate and 2/3 will accept. We assume 1/4 of the repairs will find new issues and 3/4 of the repair will not. - Gateway | *Accepted?* | *Initial Estimate Accepted*: Control tab | add Probability with value 0.67 - Gateway | *Accepted?* | *Initial Estimate Not Accepted*: Control tab | add Probability with value 0.33 - Gateway | *New Issue Found?* | *No More Issues to Repair*: Control tab | add Probability with value 0.75 - Gateway | *New Issue Found?* | *More Issues to Repair*: Control tab | add Probability with value 0.25
Condition on Sequence flows	We use Expression to return a boolean value to use as sequence flow's condition, which plays a key role in the flow's logic. - Gateway | *Have further issues?* | *More Issues to Repair*: Control tab | Condition | click "..." to open the parameter dialog, choose Expression tab | add "**{noOfIssues} != 0**" to the Expression field - Gateway | *Have further issues?* | *No More Issues to Repair*: Control tab | Condition | click "..." to open the parameter dialog, choose Expression tab | add "**{noOfIssues} = 0**" to the Expression field Note: all the outgoing transitions from a gateway should form a 100% coverage of the logic, for example, you should not use {noOfIssues} > 10 and {noOfIssues} <5 as condition expression, because values in [5, 10] will not be handled by any outgoing sequence flows.

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- On simulation, we get the following result:



Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>Customer Arrives	20
<<Activity>>Make Estimate	20
<<Gateway>>Accepted?	20
<<StartEvent>>Start Repairs	12
<<Gateway>>converge Gateway1	45
<<Activity>>Inspect for Issue	45
<<Gateway>>converge Gateway2	64
<<Gateway>>New Issue Found?	64
<<Activity>>Handle New Found Issue	19
<<Activity>>Repair Issue	45
<<Gateway>>Have further issues?	45
<<EndEvent>>Repairs Completed	12
<<Activity>>Accept Payment	12
<<EndEvent>>Customer Leaves Shop	12
<<EndEvent>>Initial Estimate Not Accepted	8

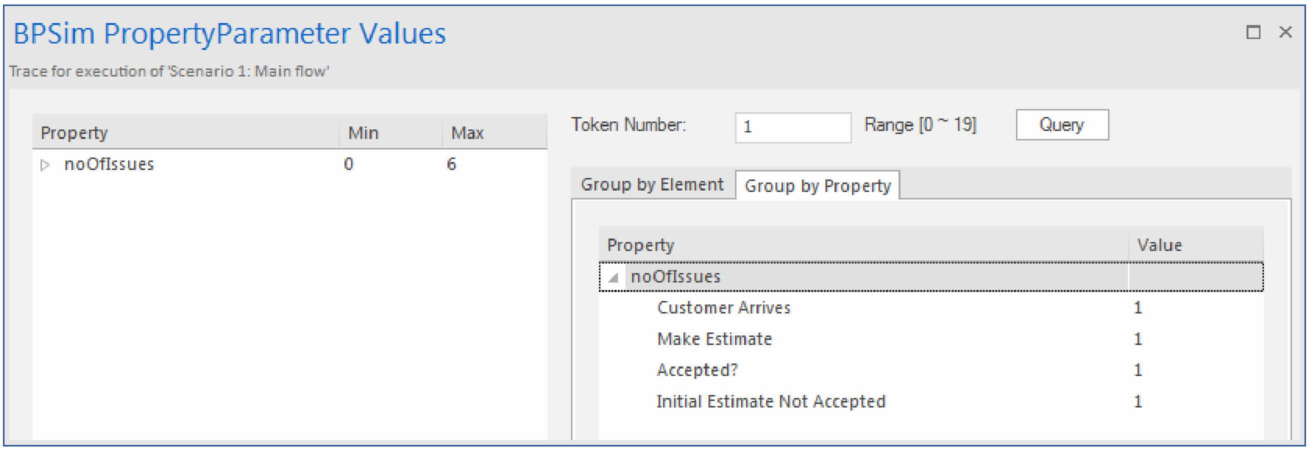
Token Analysis

- 20 customers arrives, match the number we calculated with pen and paper (refer to Configure BPSim | Customer Arrives)
- 8 customers out of 20 did not accept the initial estimate; 12 out of 20 accepted and had their car repaired. These figures match the 1/3 and 2/3 probability.
- 64 tokens passed gateway New Issue Found?, 19 of them found new issue and 45 of them did not find new issues. These figures match the 1/4 and 3/4 probability.

Analysis on each individual customer

Click the 3rd button on the toolbar to open the Property trace dialog, since there are 20 customers (tokens), you can input a number from 0 to 19 and click "Query", then do some analysis:

- This customer did not accept the initial estimate:



BPSim PropertyParameter Values

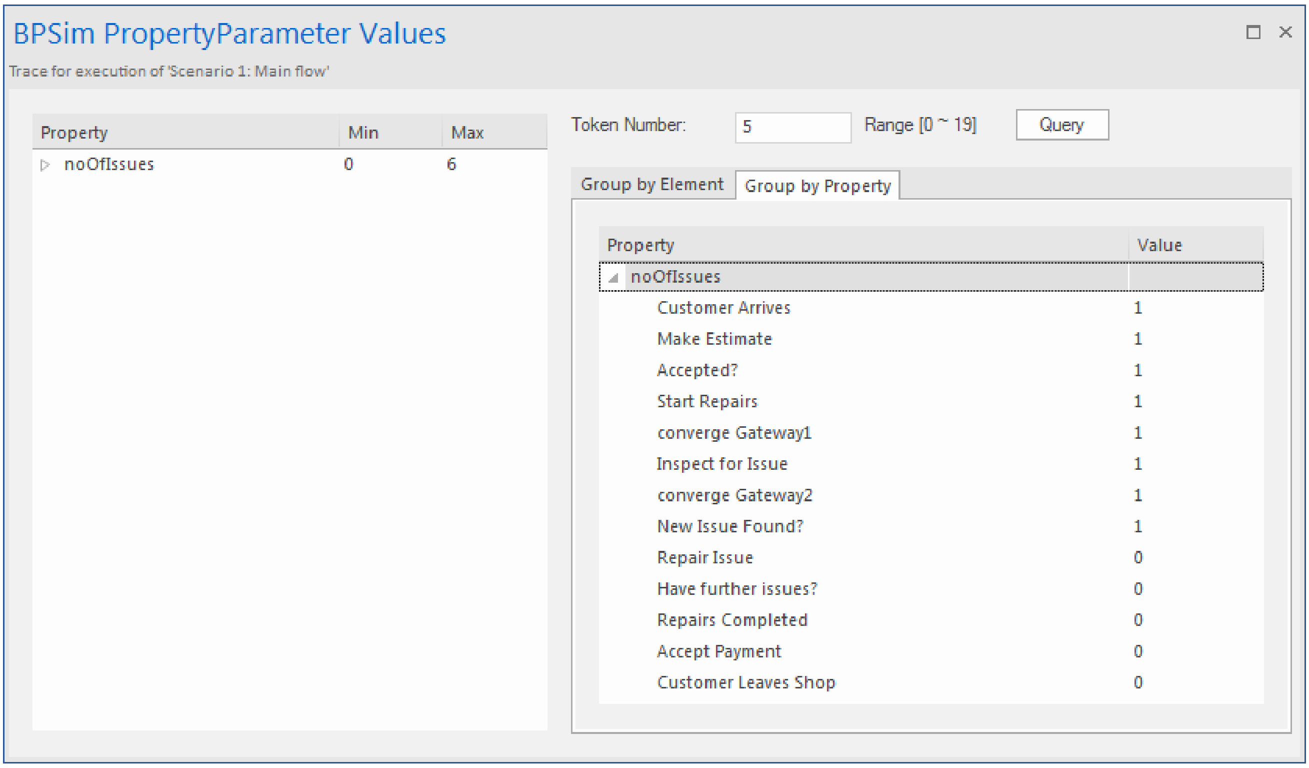
Trace for execution of 'Scenario 1: Main flow'

Property	Min	Max
noOfIssues	0	6

Token Number: 1 Range [0 ~ 19] Query

Property	Value
noOfIssues	
Customer Arrives	1
Make Estimate	1
Accepted?	1
Initial Estimate Not Accepted	1

- This customer's car has only one issue and it was fixed:



BPSim PropertyParameter Values

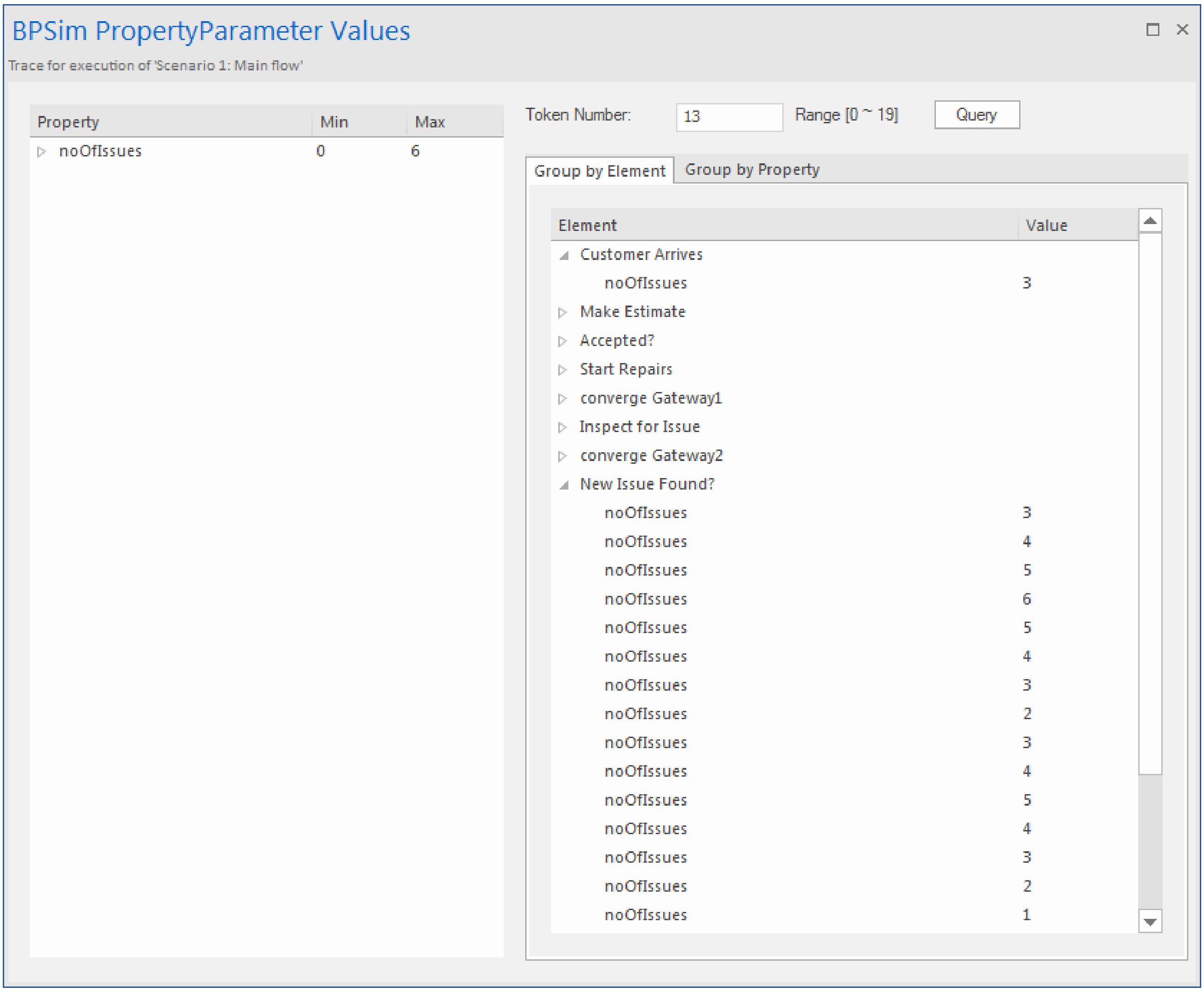
Trace for execution of 'Scenario 1: Main flow'

Property	Min	Max
noOfIssues	0	6

Token Number: 5 Range [0 ~ 19] Query

Property	Value
noOfIssues	
Customer Arrives	1
Make Estimate	1
Accepted?	1
Start Repairs	1
converge Gateway1	1
Inspect for Issue	1
converge Gateway2	1
New Issue Found?	1
Repair Issue	0
Have further issues?	0
Repairs Completed	0
Accept Payment	0
Customer Leaves Shop	0

This customer's car has 3 known issue and found 3 other issues found during repair, eventually 6 issues got fixed. (May be this is a really old car.)



BPSim PropertyParameter Values

Trace for execution of 'Scenario 1: Main flow'

Property	Min	Max
▷ noOfIssues	0	6

Token Number: 13 Range [0 ~ 19] Query

Group by Element | Group by Property

Element	Value
◢ Customer Arrives	
noOfIssues	3
▷ Make Estimate	
▷ Accepted?	
▷ Start Repairs	
▷ converge Gateway1	
▷ Inspect for Issue	
▷ converge Gateway2	
◢ New Issue Found?	
noOfIssues	3
noOfIssues	4
noOfIssues	5
noOfIssues	6
noOfIssues	5
noOfIssues	4
noOfIssues	3
noOfIssues	2
noOfIssues	3
noOfIssues	4
noOfIssues	5
noOfIssues	4
noOfIssues	3
noOfIssues	2
noOfIssues	1

BPMN2.0 Events

An Event is something that happens during the course of a Process. These Events affect the flow of the Process and usually have a cause or an impact and in general require or allow for a reaction. For example, the start of an Activity, the end of an Activity, the change of state of a document, a Message that arrives, etc., all could be considered Events.

Events allow for the description of "event-driven" Processes. In these Processes, there are three main types of Events:

23. Start Events, which indicate where a Process will start.
24. End Events, which indicate where a path of a Process will end.
25. Intermediate Events, which indicate where something happens somewhere between the start and end of a Process.

Within these three types, Events come in two flavors:

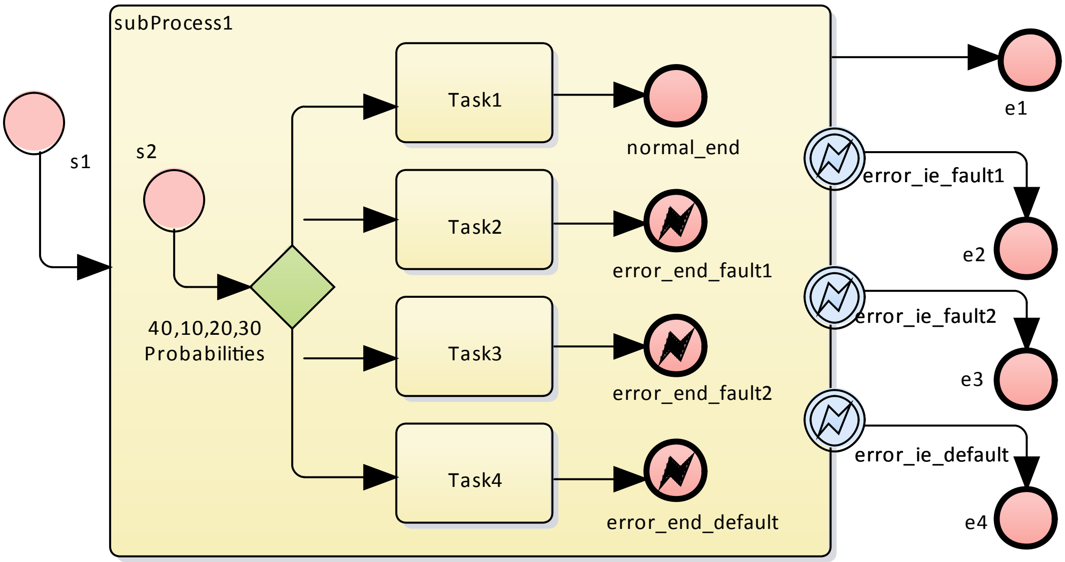
1. Events that catch a trigger. All Start Events and some Intermediate Events are catching Events.
2. Events that throw a Result. All End Events and some Intermediate Events are throwing Events that MAY eventually be caught by another Event.

In this section, we provided examples covers most of the BPMN 2.0 events that are commonly used. In each example, we provided step by step BPMN modeling, step by step BPSim Configuration and a thorough analysis on the simulation result. All the examples are available in EAExample model.

Error Event

When an intermediate Error event connects to the boundary of an activity it is used for exceptional flow. The event is triggered when a token causes a fault name to be raised in normal flow, using an Error end event.

Create BPMN Model



Create the main process

- Create a Start Event *s1*, sequence flow to
- a subProcess *subProcess1,* Enlarge it and make it expanded (Context menu | check "IsExpanded"), sequence flow to
- an End Event *e1*
- Create 3 Boundary Intermediate Events from tool box and drop on *subProcess1*, select "Edge-Mounted" | select "Error" from the menu

  - error_ie_fault1, sequence flow an EndEvent e2

  - error_ie_fault2, sequence flow an EndEvent e3

  - error_ie_default, sequence flow an EndEvent e4

Create the sub process

- Create a Start Event *s2*, sequence flow to
- a exclusive Gateway "*40,10,20,30 Probabilities*"*,* sequence flow to

  - an abstractTask *Task1*, sequence flow to an EndEvent(None) *normal_end*

  - an abstractTask *Task2*, sequence flow to an EndEvent(Error) error*_end_fault1*

  - an abstractTask *Task3*, sequence flow to an EndEvent(Error) error*_end_fault2*

  - an abstractTask *Task4*, sequence flow to an EndEvent(Error) error*_end_default*

Hookup the Error code

Create BPMN2.0::Error elements

We first create *Fault1* and *Fault2*. They will be used as error code by events.

Double click error_end_fault1 | navigate to tag "errorRef" | click "..." | browse to the package containing this model | Add New | "Fault1" OK | Add New | "Fault2" OK.

Setup Events for Error Codes:

- Double click error_end_fault1 | navigate to tag "errorRef" | click "..." | choose Fault1
- Double click error_end_fault2 | navigate to tag "errorRef" | click "..." | choose Fault2
- Double click error_ie_fault1 | navigate to tag "errorRef" | click "..." | choose Fault1
- Double click error_ie_fault2 | navigate to tag "errorRef" | click "..." | choose Fault2

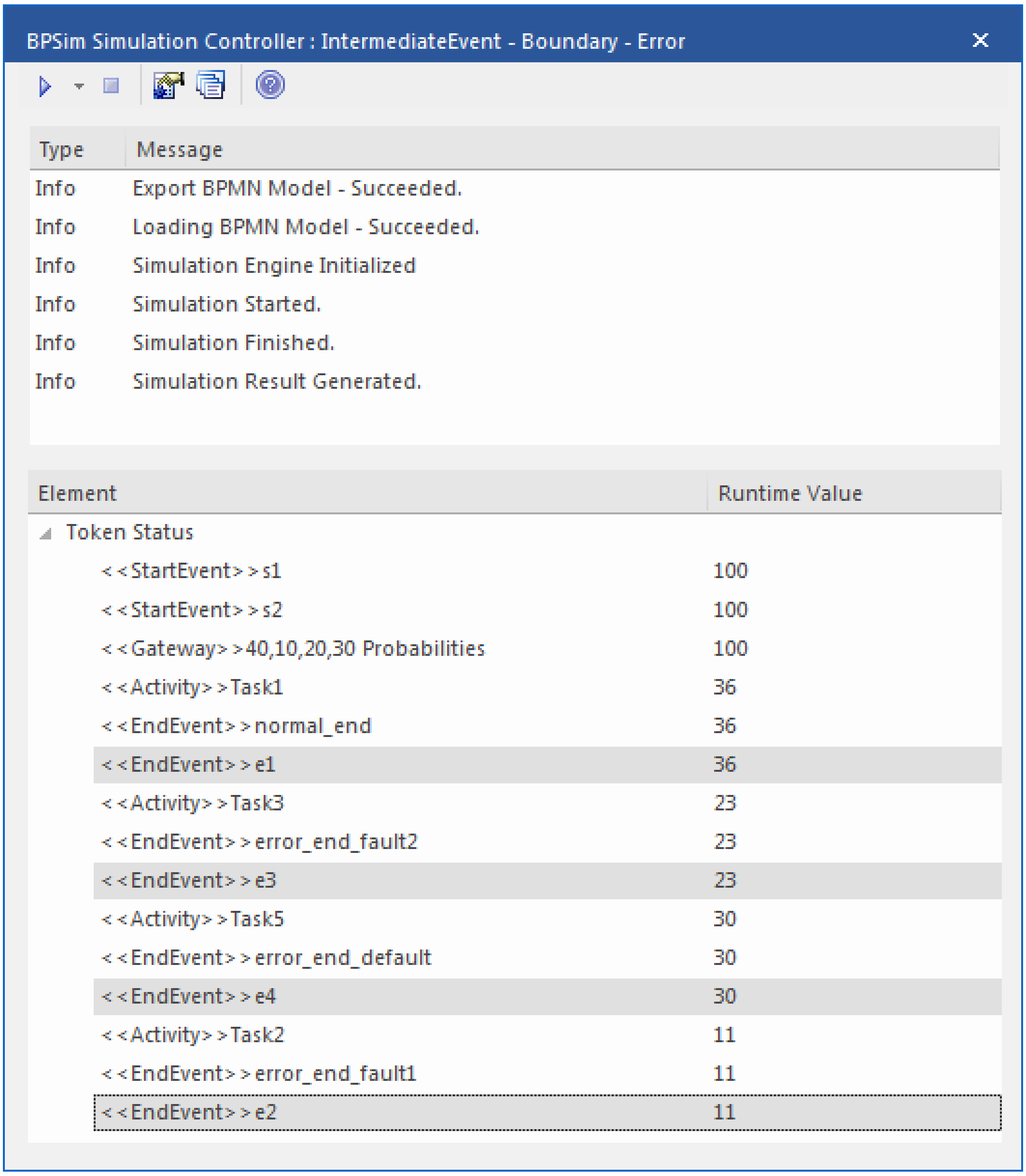
Configure BPSim

Artifact & Package	• Open BPSIm Manager • Create Artifact named "IntermediateEvent - Boundary - Error" • Select Package of the model created Then all the BPMN elements will be loaded in to the manager.
s1	• From the tree in the BPSim Manager, navigate to StartEvent | s1 • Activate *Control* Tab • Expand the drop menu on "New Parameter..." and select "TriggerCount" • Give a value of 100
Probability	From the tree in the BPSim Manager, navigate to the elements in the tree control. *Tips: You can also float the BPSim Manager view, then click the element or connectors on the BPMN diagram, the element in BPSim Manager will be automatically navigated.* • Gateway | *40,10,20,30 Probabilities* | >: Task1 | Activate *Control* Tab | Create "PRobability" | Give a value of 0.4 • Gateway | *40,10,20,30 Probabilities* | >: Task2 | Activate *Control* Tab | Create "Probability" | Give a value of 0.1 • Gateway | *40,10,20,30 Probabilities* | >: Task3 | Activate *Control* Tab | Create "Probability" | Give a value of 0.2 • Gateway | *40,10,20,30 Probabilities* | >: Task4 | Activate *Control* Tab | Create "Probability" | Give a value of 0.3

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- On simulation, we get the following result:



BPSim Simulation Controller : IntermediateEvent - Boundary - Error

Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>s1	100
<<StartEvent>>s2	100
<<Gateway>>40,10,20,30 Probabilities	100
<<Activity>>Task1	36
<<EndEvent>>normal_end	36
<<EndEvent>>e1	36
<<Activity>>Task3	23
<<EndEvent>>error_end_fault2	23
<<EndEvent>>e3	23
<<Activity>>Task5	30
<<EndEvent>>error_end_default	30
<<EndEvent>>e4	30
<<Activity>>Task2	11
<<EndEvent>>error_end_fault1	11
<<EndEvent>>e2	11

Analysis:

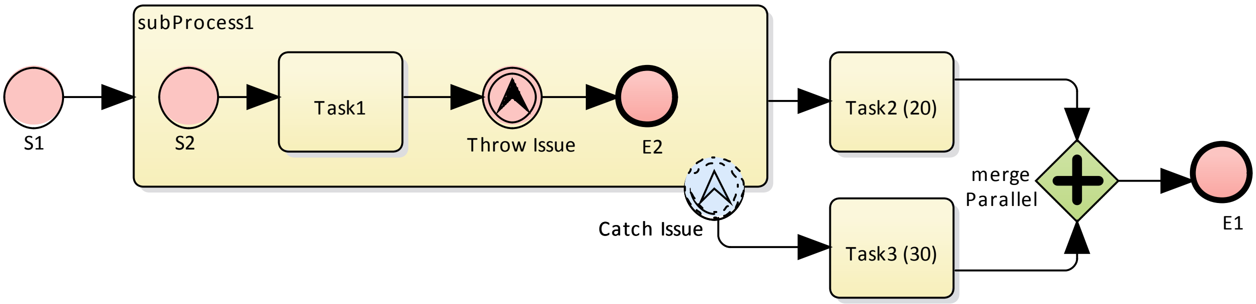
The Probability set on the sequence flows outgoing from *40,10,20,30 Probabilities* are 0.4, 0.1, 0.2 and 0.3 respectively,

- 36 out of 100 finished to *normal_end*, which will flow to *e1*
- 11 out of 100 finished to *error_end_fault1*, which will trigger *error_ie_fault1* by same ErrorRef *Fault1* and the exception flow finished at *e2*.
- 23 out of 100 finished to *error_end_fault2*, which will trigger *error_ie_fault2* by same ErrorRef *Fault2* and the exception flow finished at *e3*.
- 30 out of 100 finished to *error_end_default*, which will trigger *error_ie_default* because both of them did not set ErrorRef and the exception flow finished at *e4*.
- The number 36, 11, 23 and 30 added up to 100 which was set as TriggerCount in *s1* so they covered the 100% probability fact.

Escalation Event

In BPMN, Escalation is the non-interrupting counterpart of Error, with similar throw-catch behavior. Unlike Error, however, the normal flow and exception flow exits from the activity are not alternative paths but parallel.

Create BPMN Model



Create the main process

- Create a Start Event *S1*, sequence flow to
- a subProcess *subProcess1,* Enlarge it and make it expanded (Context menu | check "IsExpanded"), sequence flow to
- an abstract task *Task2 (20)*, sequence flow to
- a Parallel Gateway *merge Parallel*, sequence flow to
- an End Event *E1*
- Add a boundary non-interrupting Escalation Event *Catch Issue* on *subProcess1*, sequence flow to
- an abstract task *Task3 (30)*, sequence flow to *merge Parallel*

Create the sub process

- Create a Start Event *S2*, sequence flow to
- an abstract task *Task1*, sequence flow to
- an Throwing Escalation Intermediate Event *Throw Issue*, sequence flow to
- an End Event *E2*

Hookup the Escalation code

Create BPMN2.0::Escalation elements

From the toolbox, expand BPMN 2.0 Types group, drag Escalation and drop on diagram, name it Escalation1, which will be used as escalation code by events.

Setup Events for Escalation Codes:

- Double click *Throw Issue*| navigate to tag "escalationRef" | click "..." | choose *Escalation1*
- Double click *Catch Issue*| navigate to tag "escalationRef" | click "..." | choose *Escalation1*

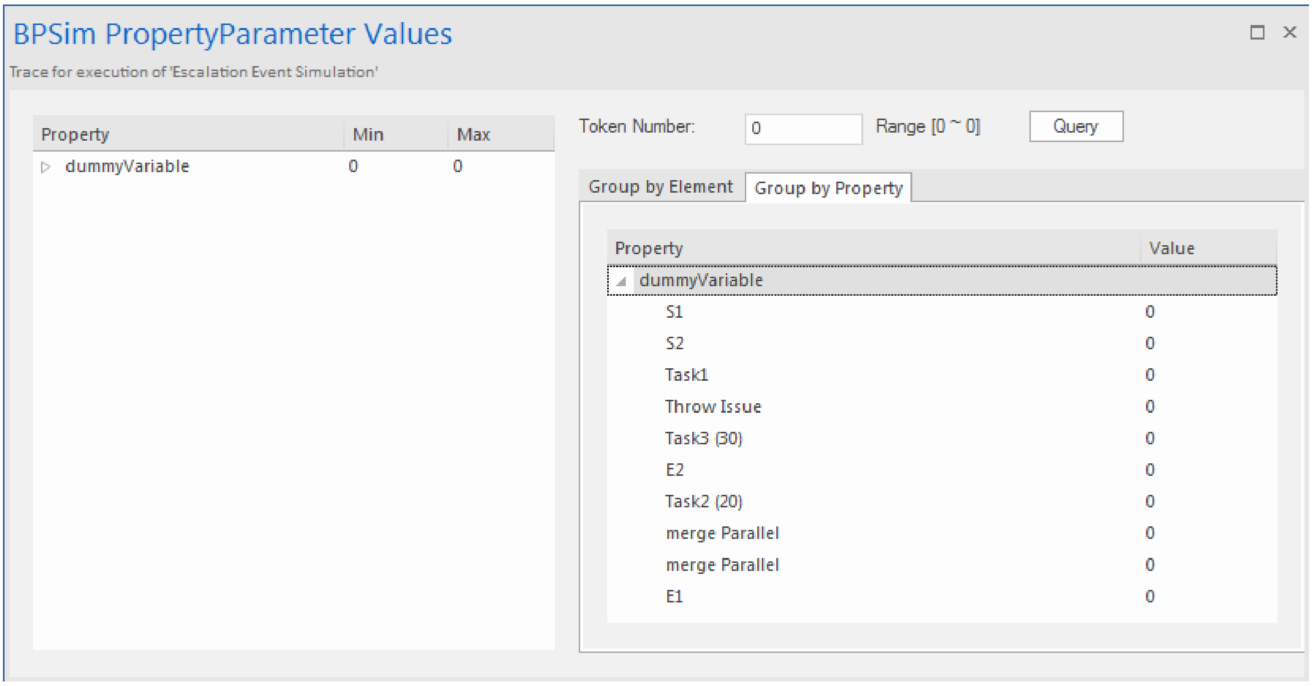
Configure BPSim

Artifact & Package	• Open BPSIm Manager • Create Artifact named "Escalation Event Simulation" • Select Package of the model created Then all the BPMN elements will be loaded in to the manager.
TriggerCount of Start Event	• From the tree in the BPSim Manager, navigate to StartEvent | S1 • Activate *Control* Tab • Expand the drop menu on "New Parameter..." and select "TriggerCount" • Give a value of 1
ProcessingTime	Set Task2 (20)'s processingTime to be 20 seconds Set Task3 (30)'s processingTime to be 30 seconds
dummyVariable for Trace	In order to show the exact trace of a given token, we need a dummy variable set on S1. select S1 | Properties Tab | input dummyVariable (or any variable name) | give it value: Numeric 0

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- After simulation, open the property trace dialog by clicking the 3rd button on the tool bar.
- click "Query", activate page "Group by Property" and expand dummyVariable



BPSim PropertyParameter Values

Trace for execution of 'Escalation Event Simulation'

Property	Min	Max
▷ dummyVariable	0	0

Token Number: 0 Range [0 ~ 0]

Property	Value
◢ dummyVariable	
S1	0
S2	0
Task1	0
Throw Issue	0
Task3 (30)	0
E2	0
Task2 (20)	0
merge Parallel	0
merge Parallel	0
E1	0

Analysis:

Unlike Error, the normal flow and exception flow exits from subProcess1 are not alternative paths but parallel. This feature can be easily discovered from the trace:

- E2 and Task2 (20) still get traversed after Task3 (30) started.
- E1 was reached after merge Parallel were traversed twice.

Event Sub-Process

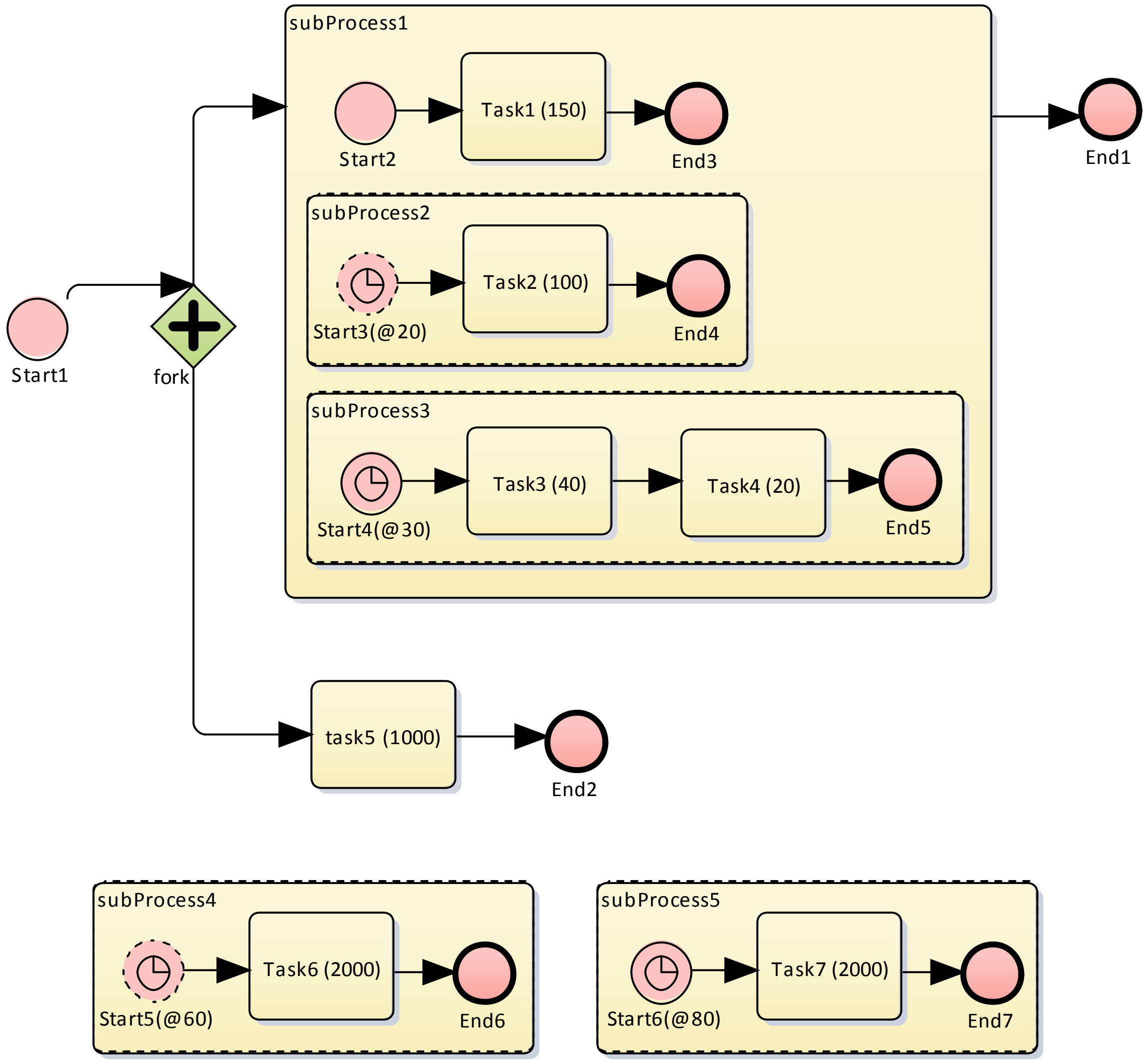
Event Sub-Processes allow to handle an Event within the context of a given Sub-Processes or Process.

Event Sub-Processes allow to handle an Event within the context of a given Sub-Processes or Process. An Event Sub-Process always begins with a Start Event. It is not instantiated by normal control flow but only when the associated Start Event is triggered. Event Sub-Processes are self-contained and MUST not be connected to the rest of the Sequence Flows in the Sub-Processes.

- If the isInterrupting attribute of its Start Event is set, an Event Sub-Process cancels execution of the enclosing Sub-Process;
- If the isInterrupting attribute is not set, execution of the enclosing Sub-Process continues in parallel to the Event Sub-Process.

In this example, we demonstrated how Interrupting and Non-Interrupting Event Sub-Processes affect the life line of the enclosing Sub Process and Process.

Create BPMN Model



Model the main process

- Create a StartEvent *Start1*, Sequence flow to

- Parallel Gateway *fork* Sequence Flow to

  - a Sub-Process *subProcess1*, sequence flow to End Event *End1*

  - an abstract Task T*ask5*, sequencd flow to End Event *End2*

Tips on how to model an Event Sub-Process.

- Drag an Activity from the BPMN2.0 toolbox and drop on the diagram
- Double Click the Activity | choose subProcess from the Type combobox |set triggeredByEvent to "true" | OK
- Context menu on the element |make the menu "Is Expanded" checked.This will show the name on the top left corner.

Model the Event Sub-Processes for the main process

- Create Event Sub-Process subProcess4

- Create a Timer Start Event *Start5(@60)* | from the Properties dialog change the type to "Event Sub-Process Non-Interrupting | Timer", sequence flow to an abstract task *Task6(2000)*, sequence flow to End Event *End6*

- Create Event Sub-Process subProcess5

- Create a Timer Start Event *Start6(@80)* | from the Properties dialog change the type to "Event Sub-Process Interrupting | Timer", sequence flow to an abstract task *Task7(2000)*, sequence flow to End Event *End7*

Model the Sub-Process subProcess1 and the enclosed Event Sub-Processes

- Create a StartEvent *Start2*, Sequence flow to an abstract task *Task1(150)*, Sequence Flow to an End Event *End3*
- Create Event Sub-Process *subProcess2*

- Create a Timer Start Event *Start3(@20)* | from the Properties dialog change the type to "Event Sub-Process Non-Interrupting | Timer", sequence flow to an abstract task *Task2(100)*, sequence flow to End Event *End4*

- Create Event Sub-Process subProcess3

- Create a Timer Start Event *Start4(@30)* | from the Properties dialog change the type to "Event Sub-Process Interrupting | Timer", sequence flow to an abstract task *Task3(40)*, sequence flow to an abstract task *Task4(20)*, Sequence Flow to an End Event *End5*

Configure BPSim

In this section, we create the Artifact, Configure package and configure the parameter values of each element.

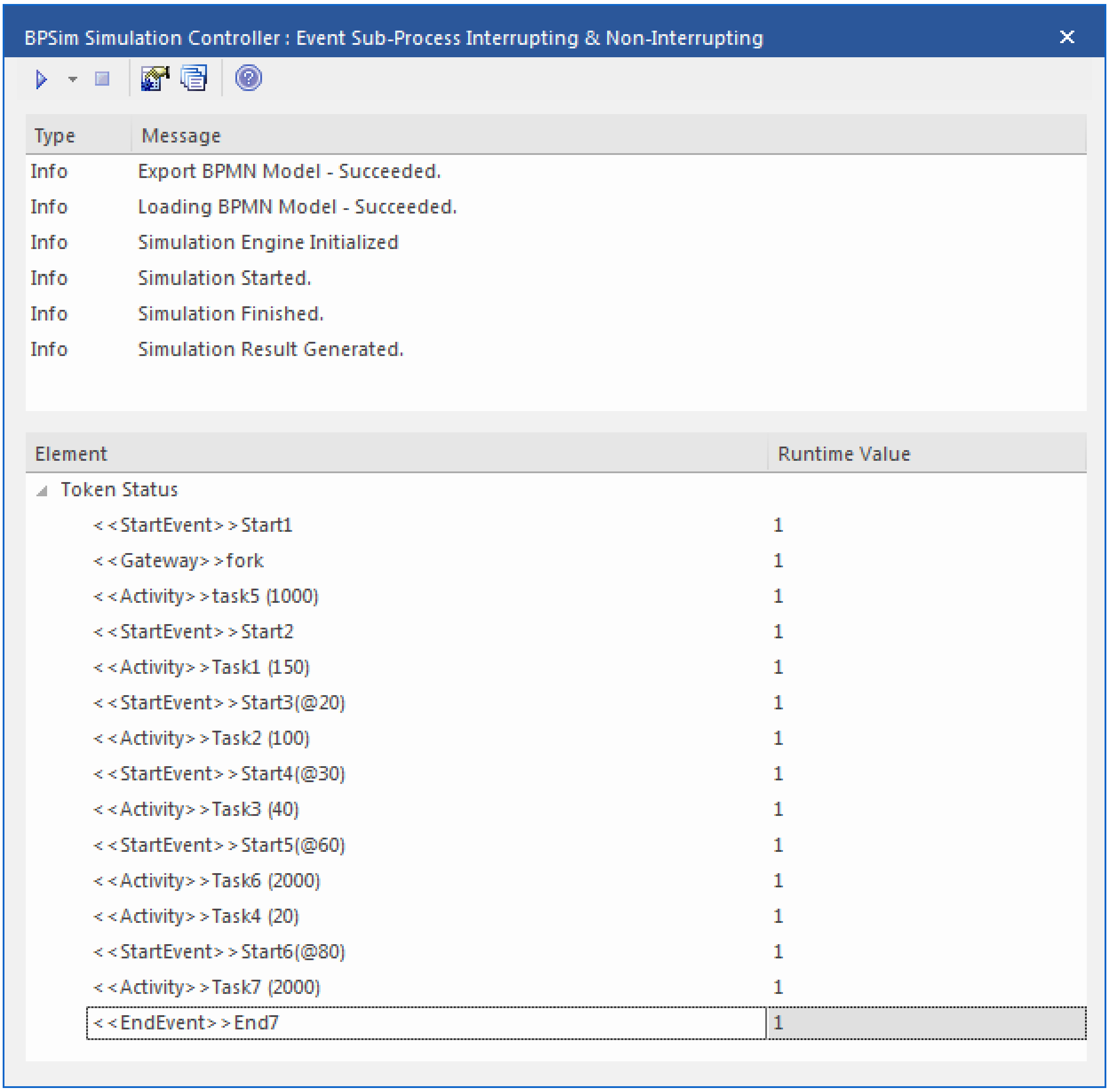
InterTriggerTimer for Start Events in Event Sub-Process	Set value for parameter Control | InterTriggerTimer for each of the following elements. Click the "..." button on the Value column to open the parameter dialog. choose Constant | Numeric, input value and select "seconds". • Start3(@20): 20 seconds • Start4(@30): 30 seconds • Start5(@60): 60 seconds • Start6(@80): 80 seconds
ProcessingTime for Tasks	Set value for parameter Time | ProcessingTime for each of the following elements. Click the "..." button on the Value column to open the parameter dialog. choose Constant | Numeric, input value and select "seconds". • Task1 (150): 150 seconds • Task2 (100): 100 seconds • Task3 (40): 40 seconds

	• Task4 (20): 20 seconds • Task5 (1000): 1000 seconds • Task6 (2000): 2000 seconds • Task7 (2000): 2000 seconds

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation | On simulation, we get the following result:



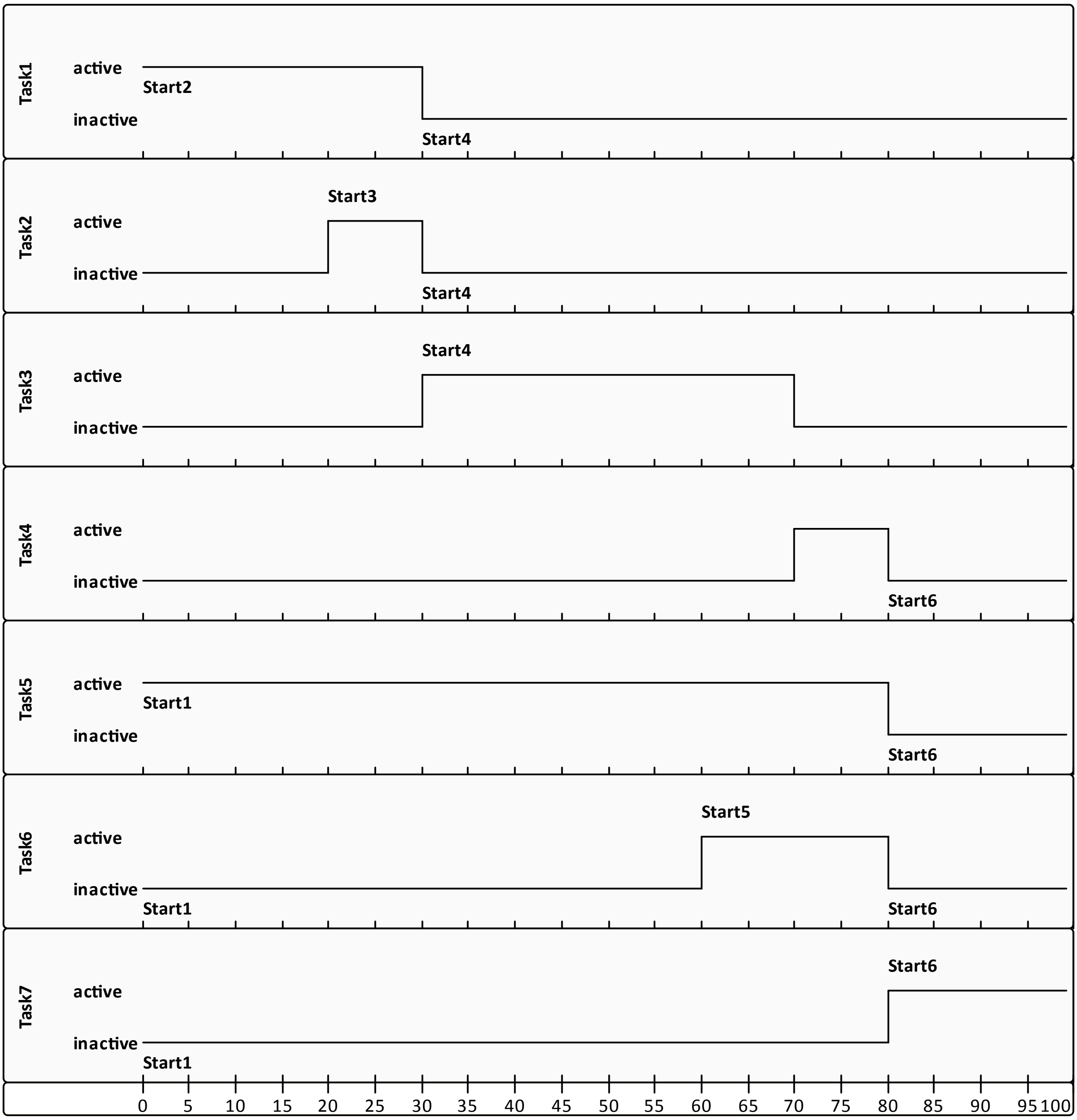
BPSim Simulation Controller : Event Sub-Process Interrupting & Non-Interrupting

Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>Start1	1
<<Gateway>>fork	1
<<Activity>>task5 (1000)	1
<<StartEvent>>Start2	1
<<Activity>>Task1 (150)	1
<<StartEvent>>Start3(@20)	1
<<Activity>>Task2 (100)	1
<<StartEvent>>Start4(@30)	1
<<Activity>>Task3 (40)	1
<<StartEvent>>Start5(@60)	1
<<Activity>>Task6 (2000)	1
<<Activity>>Task4 (20)	1
<<StartEvent>>Start6(@80)	1
<<Activity>>Task7 (2000)	1
<<EndEvent>>End7	1

Analysis:

It may not be quite straight forward to see what had happened from the above result; However, if we draw the lifeline for each task in a Timing diagram, it becomes quite clear:



- *Event Start3(@20)* is Non-Interrupting, *it did not stop Task1* @ 20 seconds
- *Event Start4(@30)* is Interrupting, it stopped *Task1* and *Task2* @ 30 seconds; it did not affected *Task5* becasue *Task5*'s enclosing process (main process) level is higher than *Start4*'s enclosing Sub-Process (*subProcess1*)
- Event *Start5(@60)* is Non-Interrupting, it started Task6 @60 seconds without affecting Task3 or Task5
- Event *Start6(@80)* is Interrupting, it started Task7 @80 seconds and interrupted the running Tasks (Task4, Task5, Task6) that were in the same or lower level of its enclosing process.
- Only *End7* is reached as expected

Fibonacci Numbers Generator with Link Event

A Link Event is a mechanism for connecting two sections of a Process. Link Events can be used:

- to create looping situations. They can be used as generic "Go To" objects within the Process level
- to avoid long Sequence Flow lines. Paired Link Events can be used as "Off-Page Connectors" for printing a Process across multiple pages.

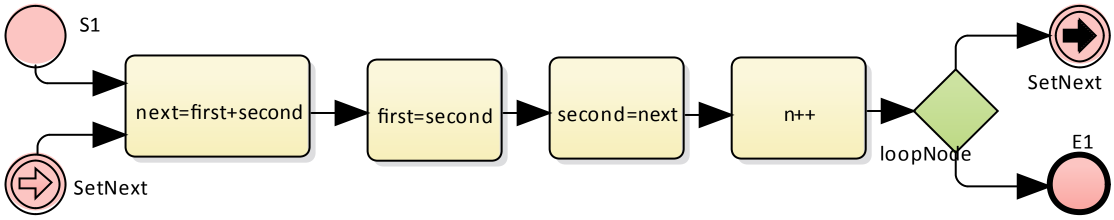
The use of Link Events is limited to a single Process level (i.e., they cannot link a parent Process with a Sub-Process).

There can be multiple source Link Events, but there can only be one target Link Event.

- The Target Link Event marker is unfilled, to "catch" from source Link
- The Source Link Event marker is filled to "throw" to the target link

When EABPSim Execution Engine is running the simulation, the source-target Link Events are paired by element **NAME**, so they can not be empty.

Create BPMN Model



- Create StartEvent *S1*, sequence flow to
- an abstract task *next=first+second*, sequence flow to
- an abstract task *first=second*, sequence flow to
- an abstract task *second=next*, sequence flow to
- an abstract task *n++*, sequence flow to
- an exclusive Gateway *loopNode,* sequence flow to
  - Throwing Link Intermediate Event *SetNext*
  - End Event *E1*
- Create Catching Link Intermediate Event *SetNext*, sequence flow to *next=first+second*

Configure BPSim

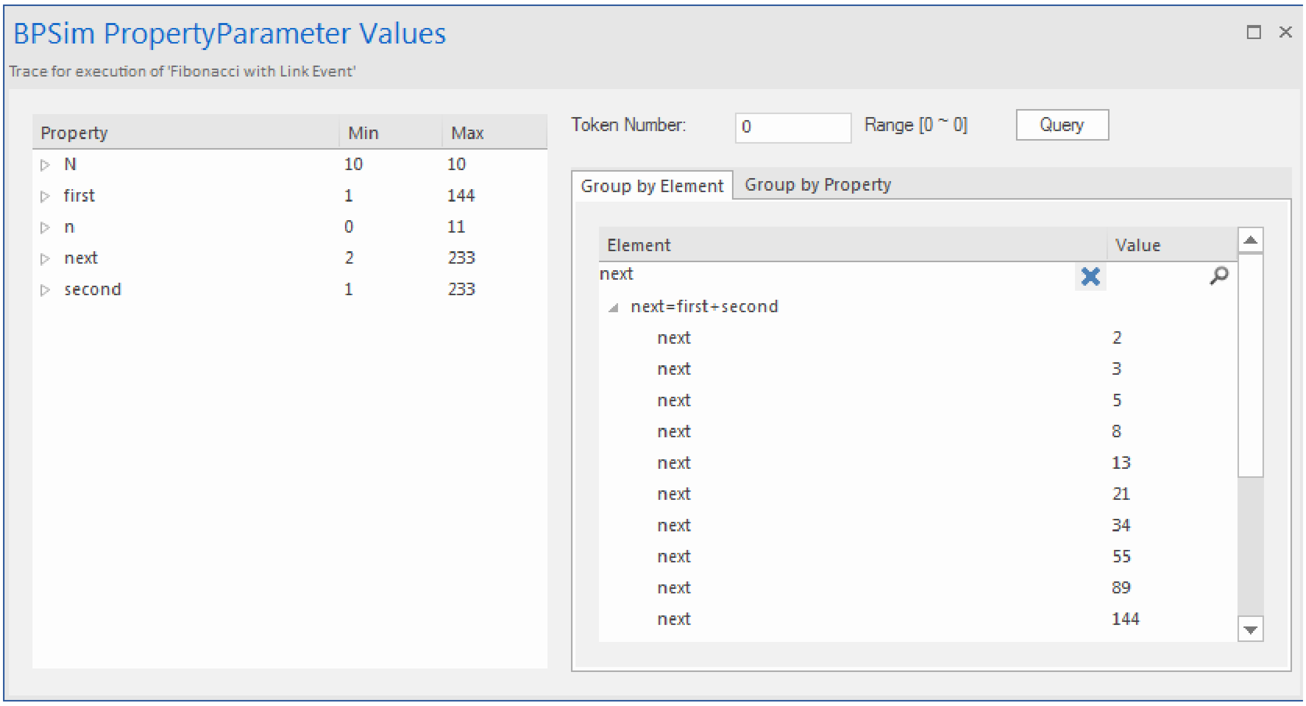
We will use Property Parameters to control the let the sequence flow form a loop during which a Fibonacci Number will be generated. The loop mechanism is implemented via the pair of Link Events.

Element: S1	Control | TriggerCount: give value "1" **Create following Properties:** • *N*: give value 10. the total Fibonacci Numbers to be generated • *first*: give value 1.

	- *second*: give value 1. - *n*: give value 0. the nth new Fibonacci number
Element: next=first+second	Create property *next* and set to Expression: {first}+{second}
Element: first=second	Set property *first* to Expression: {second}
Element: second=next	Set property *second* to Expression: {next}
Element: n++	Set property *n* to Expression: {n} + 1
Conditions of Gateway	- from Gateway to Link Intermediate Event *SetNext*: - Control | Condition | {n} <= {N} - from Gateway to End Event *E1* - Control | Condition | {n} > {N}

Run Simulation

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- After simulation, open the Property Value Trace dialog by the 3rd button on the toolbar
- Click "Query" button, activate "Group by Element" tab, expand "next=first+second"
- Now all the attributes's snapshot values are listed, apply a filter "next" and you will have:



BPSim PropertyParameter Values

Trace for execution of 'Fibonacci with Link Event'

Property	Min	Max
N	10	10
first	1	144
n	0	11
next	2	233
second	1	233

Token Number: 0 Range [0 ~ 0] Query

Group by Element | Group by Property

Element	Value
next	
next=first+second	
next	2
next	3
next	5
next	8
next	13
next	21
next	34
next	55
next	89
next	144

10 more Fibonacci numbers are generated.

1,1,2,3,5,8,13,21,34,55,89,144

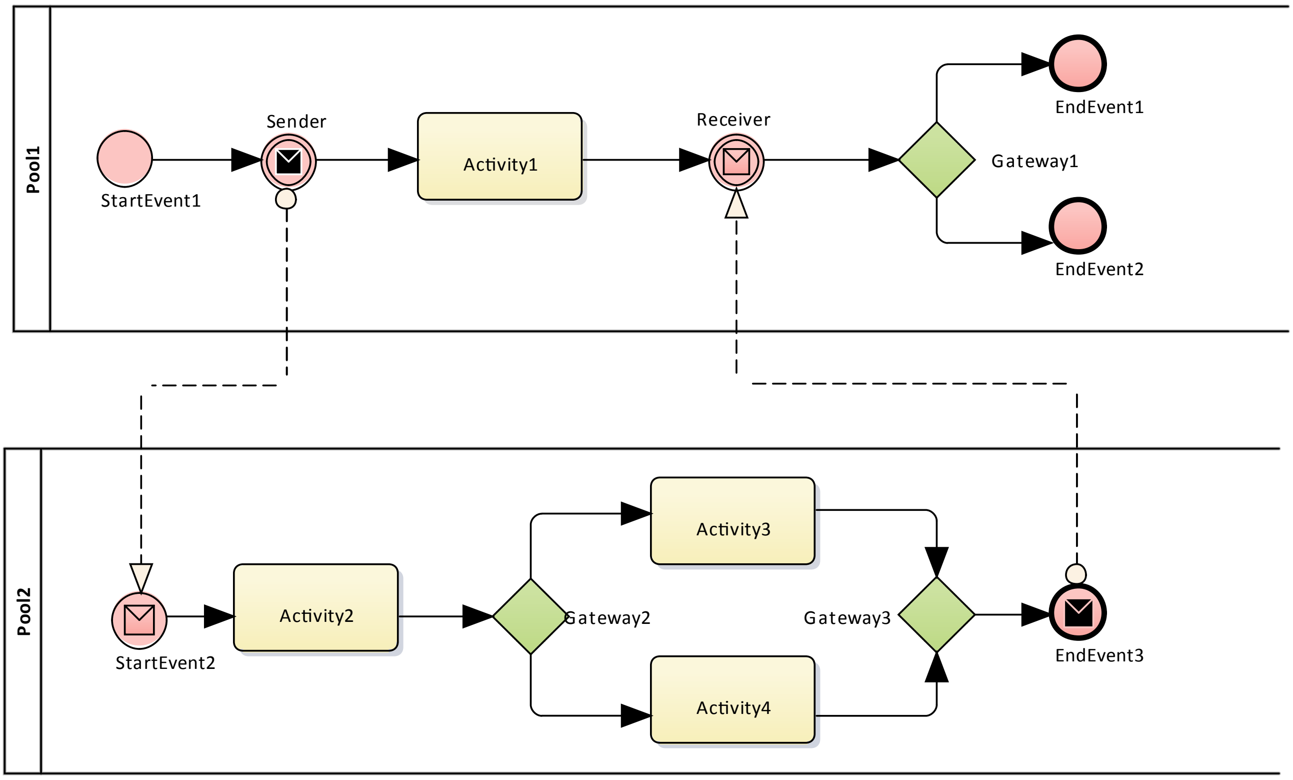
Message Event

When used in normal sequence flow the Message event may be used to send or receive a message.

- When sending a message to a participant, values of all property parameters are copied. Once the message is sent the token continues along the sequence flow.
- When receiving a message, the event is triggered when a message is received.

This example will demonstrate the features. We will first create the BPMN model, then configure BPSim and run the simulation.

Create BPMN Model



Model Explained

Pool1

- *T*oken starts from Pool1.StartEvent1
- On receiving a token, Sender (throwing Intermediate Message Event) creates a message and copy the current property values to the message.
- Sender sends the message to the "To" participant (Pool2.StartEvent2)
- Sender forwards the token along its sequence flow, as far as Receiver.
- The token will wait at Receiver until a message arrives.

Pool2

- Pool2.StartEvent2 receives a message and starts a token.
- StartEvent2 copies the values from message and set to the token.

- StartEvent2 forwards the token along its sequence flow until EndEvent3.
- EndEvent3 creates a message and copy the current property values to the message
- EndEvent3 sends the message to the “To” participant (Pool1.Receiver)

Pool1 continued

- The waiting Receiver is awaken and the property values will be updated from the arriving message.

Step by Step Creation

- Create BPMN 2.0 Collaboration diagram
- Choose the option "Create this diagram within a new CollaborationModel"
- Create *Pool1* and *Pool2*

Inside Pool1

- Create StartEvent(None) *StartEvent1*, sequence flow to
- IntermediateEvent(Throwing Message) *Sender*, sequence flow to
- Activity(abstract) *Activity1*, sequence flow to
- IntermediateEvent(Catching Message) *Receiver*, sequence flow to
- Gateway(Exclusive) *Gateway1*, sequence flows to
  - EndEvent(None) *EndEvent1*
  - EndEvent(None) *EndEvent2*

Inside Pool2

- Create StartEvent(Message) *StartEvent2*, sequence flow to
- Activity2(abstract) *Activity2*, sequence flow to
- Gateway(Exclusive) *Gateway2*, sequence flows to
  - Activity(abstract) *Activity3*
  - Activity(abstract) *Activity4*
- Merge to Gateway(Exclusive) *Gateway3*, sequence flows to
- EndEvent(Message) *EndEvent3*

Message Flows

- Message Flow from *Sender*to *StartEvent2*
- Message Flow from *EndEvent3* to *Receiver*

Configure BPSim

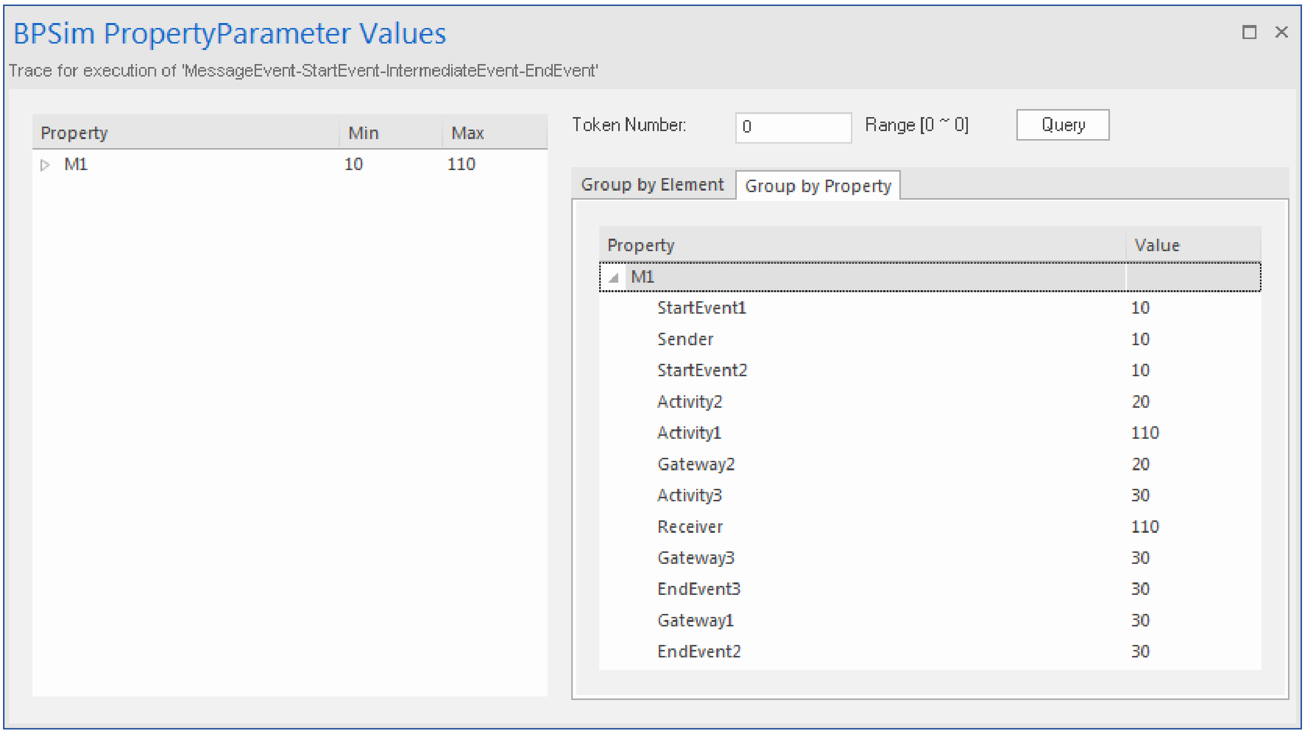
In order to show the message flow's ability of carrying values, we created a Property Parameter "M1" and change its value in each activity. Then use the value of "M1" as part of the expression for the condition of the sequence flow.

Artifact & Package	• Open BPSIm Manager • Create Artifact named "MessageEvent-StartEvent-IntermediateEvent-EndEvent" • Select Package of the model created

	Then all the BPMN elements will be loaded in to the manager.
Property Values	We will give Property Parameter "M1" initial value of 10 at the StartEvent1. Then the value is changed when the token is flowing in the processes and value copied across participants. • StartEvent1: Properties | M1, give constant numeric value "10" • Activity1: Properties | M1, give Expression value "{M1} + 100" • Activity2: Properties | M1, give Expression value "{M1} + 10" • Activity3: Properties | M1, give Expression value "{M1} + 10" • Activity4: Properties | M1, give Expression value "{M1} + 1" *Tips: the format of* {PropertyName} *is a convenient short form of* getProperty("PropertyName").
Control Parameters	We only need 1 token in this simulation for evaluating the behavior of this model, so we give "1" to StartEvent1's TriggerCount in the Control page. Now we set up the conditions for Gateway's outgoing Sequence Flows. • from "Gateway1" to "EndEvent1", Condition: {M1} >= 50 • from "Gateway1" to "EndEvent2",Condition: {M1} < 50 • from "Gateway2" to "Activity3", Condition: {M1} >= 15 • from "Gateway2" to "Activity4", Condition: {M1} < 15

Run Simulation

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- When Simulation finished, click the 3rd button to open the "Property Parameter Values" dialog, which traces the values for properties during simulation
- Token Number, input "0", and click "Query" button, Activate the "Group by Property" tab



BPSim PropertyParameter Values

Trace for execution of 'MessageEvent-StartEvent-IntermediateEvent-EndEvent'

Property	Min	Max
M1	10	110

Token Number: 0 Range [0 ~ 0]

Property	Value
M1	
StartEvent1	10
Sender	10
StartEvent2	10
Activity2	20
Activity1	110
Gateway2	20
Activity3	30
Receiver	110
Gateway3	30
EndEvent3	30
Gateway1	30
EndEvent2	30

Analysis:

Since the ProcessingTime of Activity1 was set as a distribution value, it turns out that:

- [Process1] M1's value after Pool1.StartEvent1 is "10" as expected
- [Process2] Pool2.StartEvent2 has M1 as 10, this value is carried from a message sent by Pool1.Sender

Now there are actually two M1, Process1.M1 and Process2.M1

- [Process2] Pool2.Activity2 increased Process2.M1 by 10. [Process2.M1 == 20]
- [Process1] Pool1.Activity1 increased Process1.M1 by 100. [Process1.M1 == 110]
- [Process2] The condition expression are evaluated, since "20 > 15", the token will flow to Activity3 [Process2.M1 == 20]
- [Process2] Pool2.Activity3 increased Process2.M1 by 10. [Process2.M1 == 30]
- [Process1] Pool1.Receiver is reached and waiting [Process1.M1 == 110]
- [Process2] Pool2.Gateway3 serves as a merge node and continue to EndEvent3 [Process2.M1 == 30]
- [Process1] Pool1.Receiver woke up by message(carrying M1 == 30) and Process1.M1's value changed from 110 to 30
- [Process1] The condition expression are evaluated, since "30 < 50", the token will flow to EndEvent2 [Process1.M1 == 30]

Note:

- The lines in blue are the effects of message flows.
- The order in a process is defined; however, the order between two processes are not always predictable.
- The Throwing Message Event fork another process; the Catching Message serves like thread synchronization

Signal Events

A signal event has the advantage of loosely coupling the thrower and catchers by publish-subscribe integration.

The thrower will broadcast the signal rather than addressing it to a particular process; any process listening for that particular event could trigger a new instance using a signal start event.

Signal can be thrown from either a throwing intermediate event or end event, and may be caught in a start event, catching intermediate event (including boundary signal event).

In this example, we demonstrated the following signal events and their impact on the life line of the tasks via BPSim Parameter setting.

- start signal event:

  - *Start By Signal1* in top level process (Pool1)

  - *Start By Signal2* Interrupting in event sub-process *eventSubProcess2*

  - *Start By Signal1* Non Interrupting in event sub-process *eventSubprocess1*
- throwing intermediate signal event

  - *Broadcast Signal1*
- catching intermediate signal event:

  - *Receive Signal1(normal)*

  - *Receive Signal2(normal)*

  - *Receive Signal2(boundary Interrupting)*

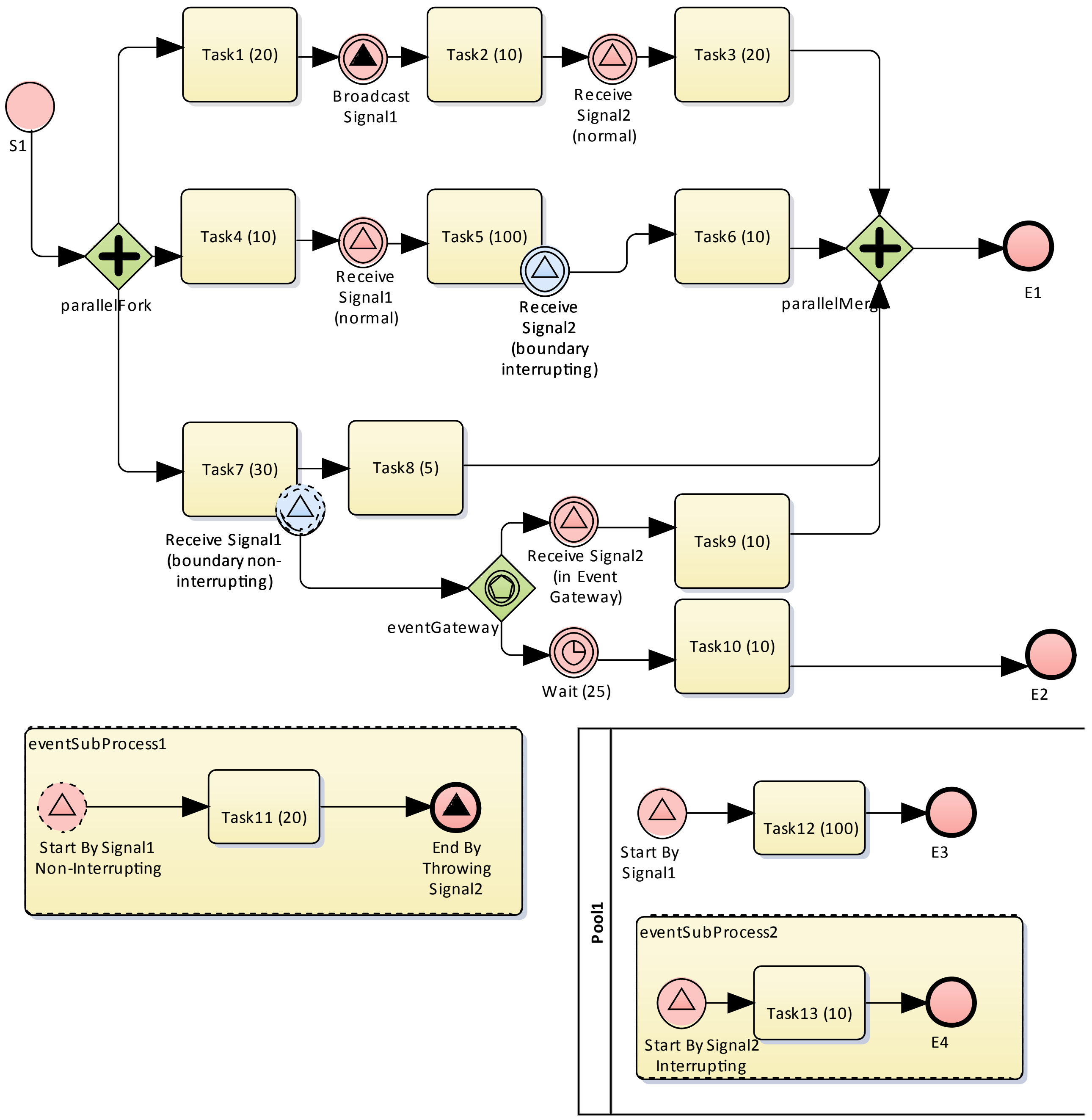
  - *Receive Signal1(boundary non-interrupting)*

  - *Receive Signal2 (in Event Gateway)*
- end signal event:

  - *End By Throwing Signal2*

Create BPMN Model

In order to demonstrate the ability to communicate across processes via signal event, we created a collaboration model with a main pool and a process in another pool (Pool1)



Create the Collaboration and main process

Create a new BPMN2.0 Collaboration diagram named *CollaborationForTestingSignalEvents*, (choose the option "Create this diagram within a new Collaboration Model"); name

Select diagram *CollaborationForTestingSignalEvents* in project browser | context menu | Encapsulate Process

Then a pool *PoolMain* and a process *BusinessProcess_PoolMain* are created and following tags are set with values:

CollaborationForTestingSignalEvents.mainPool is set to *PoolMain*

PoolMain.processRef is set to *BusinessProcess_PoolMain*

Create the elements for the main process

Start *S1* sequence flow to a fork parallel gateway *parallelFork*, sequence flows to:

- abstract task *Task1 (20)*, sequence flow to Throwing Intermediate Signal Event *Broadcast Signal1*, sequence flow to abstract task *Task2 (10)*, sequence flow to Catching Intermediate Signal Event *Receive Signal2*, sequence flow to abstract task *Task3 (20)*, sequence flow to a merge parallel gateway *parallelMerge*, sequence flow to End event *E1*
- abstract task *Task4 (10)*, sequence flow to Catching Intermediate Signal Event *Receive Signal1 (normal)*, sequence flow to abstract task *Task5 (100)* on which we create a boundary interrupting Catching Intermediate Signal Event *Receive Signal2 (boundary interrupting)*, sequence flow to abstract task *Task6 (10)*, sequence flow to *parallelMerge*
- abstract task *Task7 (30)*, sequence flow to abstract task *Task8 (5)*, sequence flow to *parallelMerge*. On *Task7 (30)*, we create a boundary non-interrupting Catching Intermediate Signal Event *Receive Signal1 (boundary non-interrupting)*, sequence flow to an Event Gateway *eventGateway*, sequence flows to:

- a Catching Intermediate Signal Event *Receive Signal2 (in Event Gateway)*, sequence flow to an abstract task *Task9 (10)*, sequence flow to *parallelMerge*

- a Catching Intermediate Timer Event *Wait (25)*, sequence flow to abstract *Task10 (10)*, sequence flow to End Event *E2*

Create an Event sub-process(triggered by non-interrupting start signal event) in the main process

- create an activity *eventSubProcess1*, set its type to *subProcess* from Property dialog, change attribute triggeredByEvent to *true*
- create a start event *Start By Signal1 Non Interrupting* in *eventSubProcess1*, set its type to *Event Sub-Process Non-Interrupting | Signal* from the Property dialog, sequence flow to
- an abstract task *Task11 (20)*, sequence flow to
- an end event *End By Throwing Signal2*, set its *type* to *Signal* from the Property dialog

Create another process

From the Toolbox, drag a pool and drop on the diagram and name it "Pool1". Select Pool1 in project browser | context menu | Encapsulate Process

Then a process *BusinessProcess_Pool1* is created and Pool1.processRef is set to *BusinessProcess_Pool1*

Create the main process for *Pool1*

- create a Signal Start Event Start By Signal1 , sequence flow to
- an abstract task Task12 (100), sequence flow to
- an end event E3

Create an Event sub-process to interrupt Pool1

- create an activity *eventSubProcess1*, set its type to *subProcess* from Property dialog, change attribute triggeredByEvent to *true*
- create a start event *Start By Signal2 Interrupting* in *eventSubProcess2*, set its type to *Event Sub-Process Interrupting | Signal* from the Property dialog, sequence flow to
- an abstract task *Task13 (10)*, sequence flow to
- an end event *E4*

Create the BPMN2.0 Signal Elements and configure for Signal Events

From the toolbox, expand BPMN 2.0 - Types group, drag Signal on the diagram, name it Signal1; do the same to create Signal2; They are root elements (which can be used by all processes) so they will be created directly under the package .

Double click the Signal Event | click "..." for signalRef tag | browse to the signal element for the following events (Tips: you can alternatively drag the created signal element from the project browser and drop on the event elements on diagram, choose the context menu "set signalRef")

- Signal1 on:
  - *Broadcast Signal1*
  - *Start By Signal1* in top level process (Pool1)
  - *Start By Signal1* Non Interrupting in event sub-process *eventSubprocess1*
  - *Receive Signal1(normal)*
  - *Receive Signal1(boundary non-interrupting)*
- Signal2 on
  - *Start By Signal2* Interrupting in event sub-process *eventSubProcess2*
  - *Receive Signal2(normal)*
  - *Receive Signal2(boundary Interrupting)*
  - *Receive Signal2 (in Event Gateway)*

Configure BPSim

In this section, we create the Artifact, Configure package and configure the parameter values of each element.

The configuration is quite simple because all signal events does not need any BPSim configurations. All we need is to set the processing time for tasks so we can observe how processes/threads/tasks are started and interrupted.

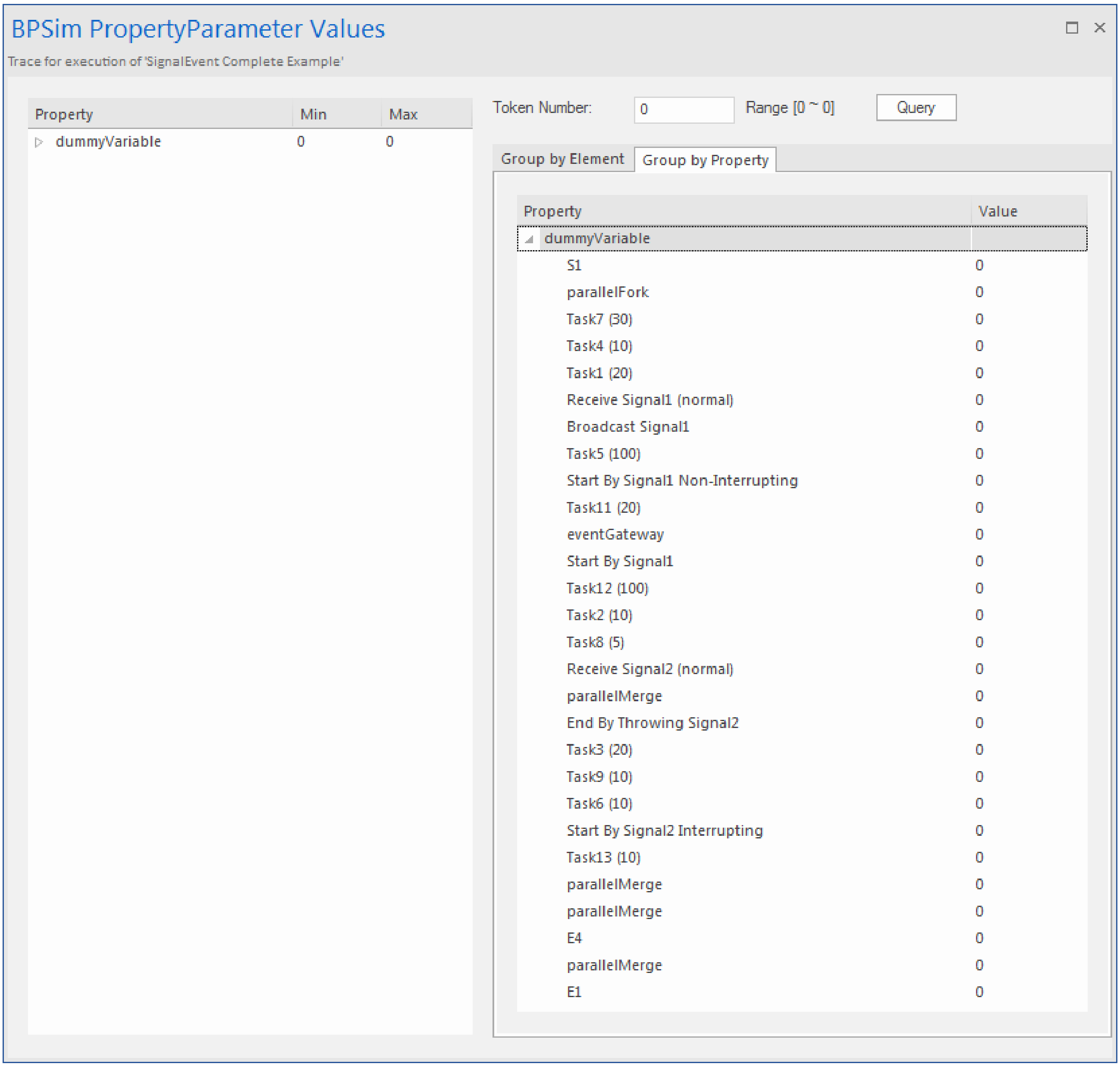
Non-Signal Events	• S1: Control | TriggerCount: give a value "1" • Wait (25): Control | InterTriggerTimer: Click the "..." button on the value column | choose Constant | Numeric | input value "25" and select "seconds" for time unit
Dummy variable for Process	The simulation controller has a list to show the runtime token count for each element. For example, 4 tokens passed element parallelMerge in this simulation. It is quite useful for certain statistics and analysis. However, it does not tell WHEN parallelMerge was traversed during the simulation. In order to get the exact trace for a single token, we need the property trace utility, which rely on property parameters. So we create a dummy one. In the Configuration manager, expand Business Process, • select "BusinessProcess_Main" | Properties tab | input dummyVariable in the list, use "..." to open the dialog | Numeric | input "0" • select "BusinessProcess_Pool1" | Properties tab | input dummyVariable in the list, use "..." to open the dialog | Numeric | input "0"
Processing Time for Tasks	Set value for parameter Time | ProcessingTime for each of the following elements. Click the "..." button on the Value column to open the parameter dialog. choose Constant | Numeric, input value and select "seconds". • Task1 (20): 20 seconds • Task2 (10): 10 seconds • Task3 (20): 20 seconds • Task4 (10): 10 seconds • Task5 (100): 100 seconds • Task6 (10): 10 seconds

	• Task7 (30): 30 seconds • Task8 (5): 5 seconds • Task9 (10): 10 seconds • Task10 (10): 10 seconds • Task11 (20): 20 seconds • Task12 (100): 100 seconds • Task13 (10): 10 seconds

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation | On simulation } Click the Attribute button (3rd on the toolbar):
- Click "Query" button | activate "Group by Property" tab | expand dummyVariable



BPSim PropertyParameter Values
Trace for execution of 'SignalEvent Complete Example'
Token Number:
Range [0 ~ 0]
Query
Group by Element
Group by Property

Property	Min	Max
dummyVariable	0	0

Token Number: 0

Property	Value
dummyVariable	
S1	0
parallelFork	0
Task7 (30)	0
Task4 (10)	0
Task1 (20)	0
Receive Signal1 (normal)	0
Broadcast Signal1	0
Task5 (100)	0
Start By Signal1 Non-Interrupting	0
Task11 (20)	0
eventGateway	0
Start By Signal1	0
Task12 (100)	0
Task2 (10)	0
Task8 (5)	0
Receive Signal2 (normal)	0
parallelMerge	0
End By Throwing Signal2	0
Task3 (20)	0
Task9 (10)	0
Task6 (10)	0
Start By Signal2 Interrupting	0
Task13 (10)	0
parallelMerge	0
parallelMerge	0
E4	0
parallelMerge	0
E1	0

Analysis:

It may not be quite straight forward to see what had happened from the above result; However, if we draw the lifeline for each task, it becomes quite clear:

- *Task1, Task4, Task7 Started parallelly*
- *Task2* started immediately after *Task1* finish (without stop at the throwing event)
- @20 seconds, *Signal1* was broadcasted by Throwing Intermediate Event *Broadcast Signal1*,

  - *Receive Signal1 (normal)* was activated and *Task5* Started

  - *Start By Signal1 Non-Interrupting* was activated and *Task11* in *eventSubProcess1* Started

  - *Start By Signal1* was activated and *Task12* in *Pool1* Started
- @40 seconds, *Signal2* was broadcasted by End Event *End By Throwing Signal2*

  - *Receive Signal2 (normal)* was activated and *Task3* Started

  - *Task5* was interrupted and *Task6* Started

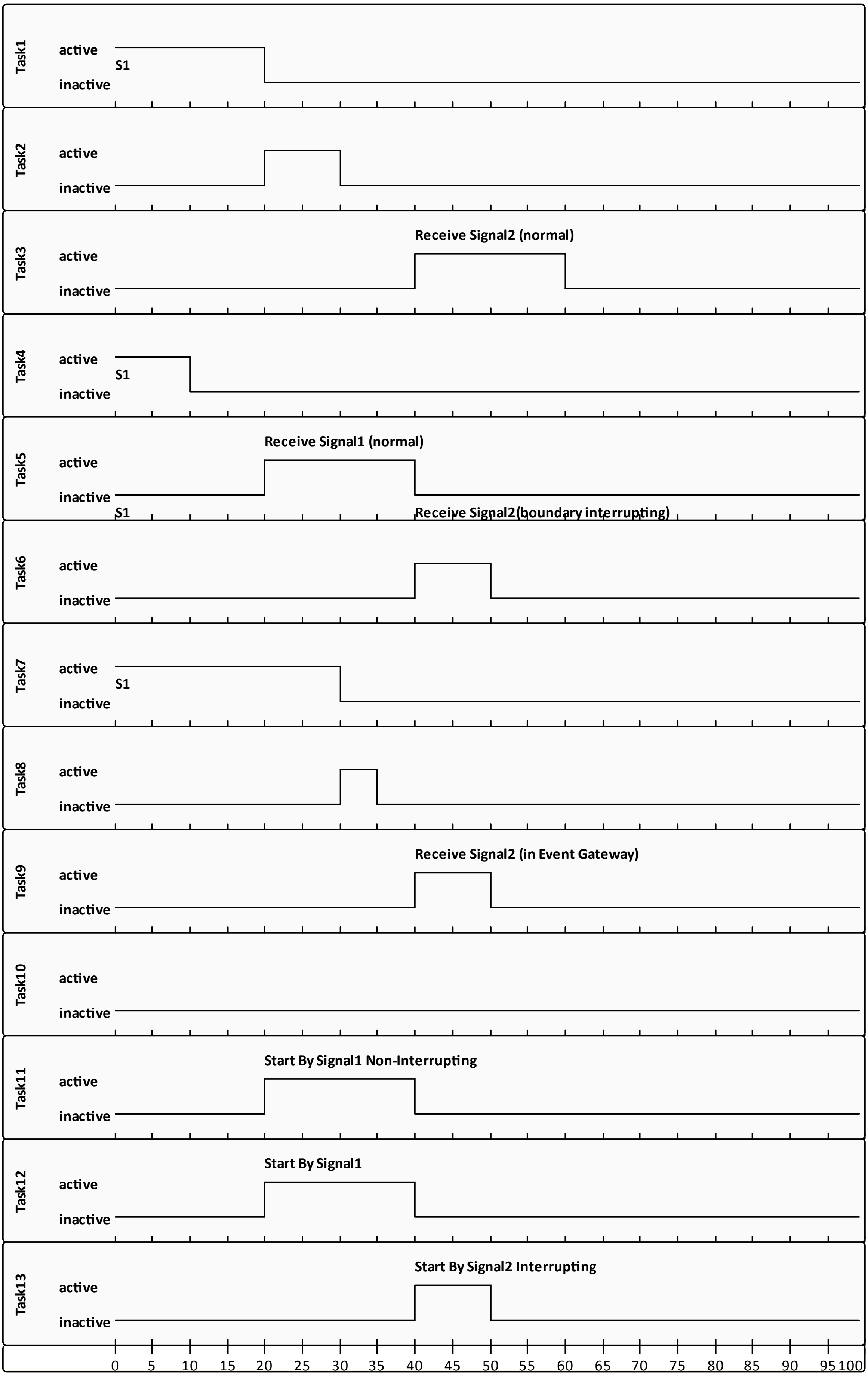
- *Receive Signal2 (in Event Gateway)* was activated and *Task9* Started

- *Start By Signal2 Interrupting* was activated, and:

  - *the main process in Pool1 was interrupted and Task12 stops*

  - *Task13* in *eventSubProcess2* Started

- The *eventSubProcess2* inside *BusinessProcess_Pool1* finished when *E4* was reached @50 seconds
- The *BusinessProcess_MainPool* finished when *E1* was reached @60 seconds
- The Intermediate Timer Event *Wait (25)* did not get activated because the signal event in the gateway was activated first; as a result, *Task10* was never started


Task1
active
S1
inactive
Task2
active
inactive
Task3
Receive Signal2 (normal)
active
inactive
Task4
active
S1
inactive
Task5
Receive Signal1 (normal)
active
inactive
S1
Receive Signal2(boundary interrupting)
Task6
active
inactive
Task7
active
S1
inactive
Task8
active
inactive
Task9
Receive Signal2 (in Event Gateway)
active
inactive
Task10
active
inactive
Task11
Start By Signal1 Non-Interrupting
active
inactive
Task12
Start By Signal1
active
inactive
Task13
Start By Signal2 Interrupting
active
inactive
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

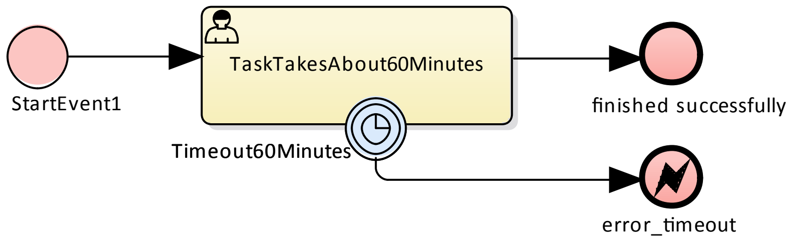
Note: The actual time running for each task can be observed from the generated BPSimReport element, by:

Double click the <<BPSimReport>> element | Expand the "Time" group | Expand the task element | Total Time In Task.

For example, for element *Task5 (100)*, although we set its processingTime to be 100 seconds, the **Total Time In Task** was 20 Seconds, which was interrupted by *Receive Signal2 (boundary interrupting)* @ 20 seconds.

Timer Event - Boundary

Create BPMN Model



- Create a Start Event *StartEvent1*, sequence flow to
- a userTask *TaskTakesAbout60Minutes*, sequence flow to
- an End Event *finished sucessfully*
- Create a Intermediate Event *Timeout60Minutes* from tool box and drop on *TaskTakesAbout60Minutes*, select "Edge-Mounted" | select "Timer" from the menu, sequence flow to
- an End Event (Error) error*_timeout*

Configure BPSim

In this section, we create the Artifact, Configure package and configure the parameter values of each element.

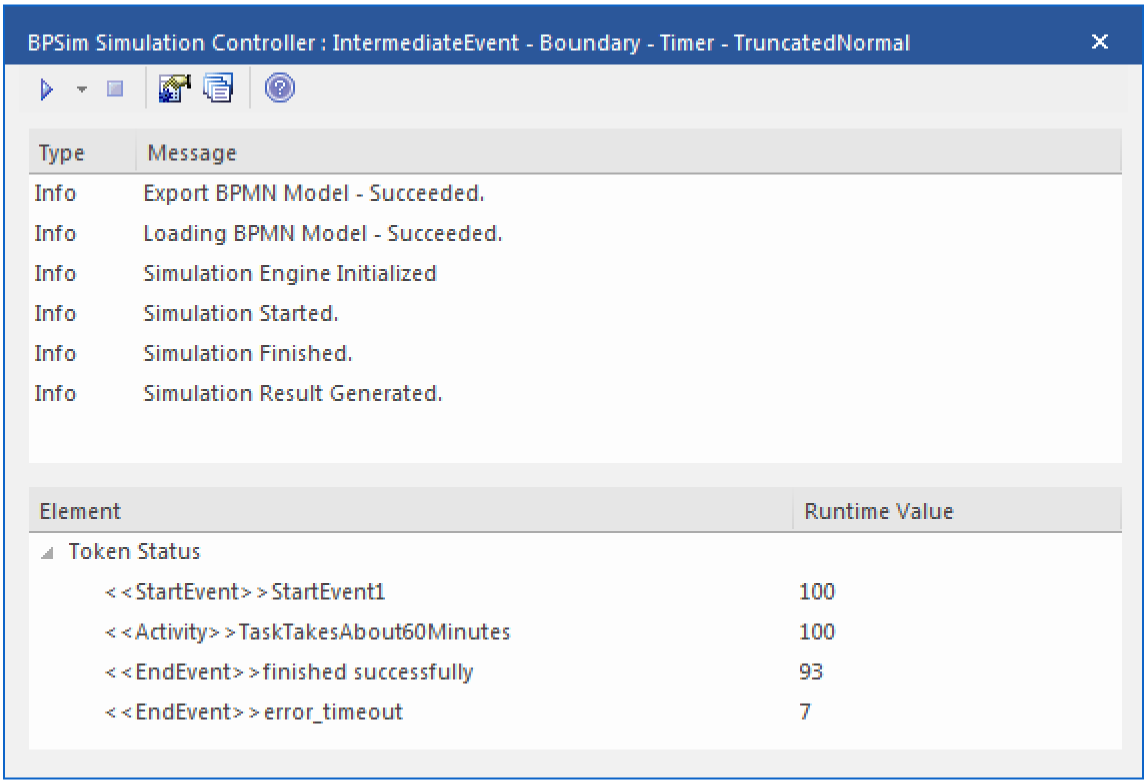
Artifact & Package	• Open BPSIm Manager • Create Artifact named "IntermediateEvent - Boundary - Timer - TruncatedNormal" • Select Package of the model created Then all the BPMN elements will be loaded in to the manager.
StartEvent1	• From the tree in the BPSim Manager, navigate to StartEvent | StartEvent1 • Activate *Control* Tab • Expand the drop menu on "New Parameter..." and select "TriggerCount" • Give a value of 100
TaskTakesAbout60Minutes	• From the tree in the BPSim Manager, navigate to Activity | TaskTakesAbout60Minutes • Activate *Time* Tab • Expand the drop menu on "New Parameter..." and select "ProcessingTime" • Click "..." button for value | Distribution | TruncatedNormal | mean: 50, standardDeviation: 10, min: 0, max: 1000 | OK
Timeout60Minutes	• From the tree in the BPSim Manager, navigate to IntermediateEvent | Timeout60Minutes • Activate *Control* Tab • Expand the drop menu on "New Parameter..." and select "InterTriggerTimer"

	• Give the value 000:000:000 001:00:00, which means 1 hour

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- On simulation, we get the following result:


BPSim Simulation Controller : IntermediateEvent - Boundary - Timer - TruncatedNormal

Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>StartEvent1	100
<<Activity>>TaskTakesAbout60Minutes	100
<<EndEvent>>finished successfully	93
<<EndEvent>>error_timeout	7

Analysis:

Since the ProcessingTime of TaskTakesAbout60Minutes was set as a distribution value, it turns out that:

- 93 out of 100 finished in 1 hour; so the normal flow to *finished successfully* takes effect.
- 7 out of 100 finished more than 1 hour; so the exception flow to *error_timeout* takes effect.

Other Configurations:

In the example folder, there are two other BPSim artifact, they set the ProcessingTime as constant value of 50 minutes and 80 minutes, other settings remains the same.

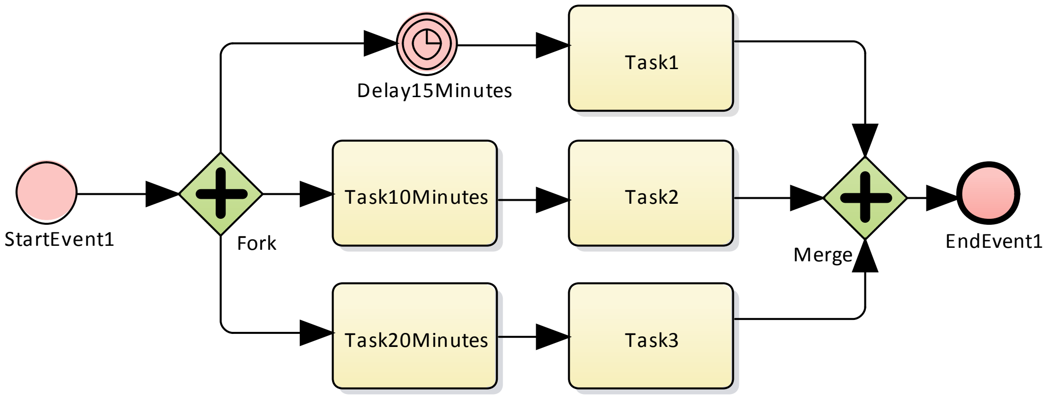
Run simulation on these two artifact:

- ProcessingTime configured to be 50 minutes always finish in normal flow
- ProcessingTime configured to be 80 minutes always finish in exception flow

Standalone IntermediateEvent - Timer

When Timer Intermediate Event is used within the normal sequence flow as a standalone element, it acts as a delay mechanism.

Create BPMN Model



- Create a Start Event *StartEvent1*, sequence flow to
- a Parallel Gateway *Fork*, sequence flow to

  - A Standalone Timer Intermediate *Delay15Minutes*, sequence flow to an Activity *Task1*

  - An Activity *Task10Minutes*, sequence flow to an Activity *Task2*

  - An Activity *Task20Minutes*, sequence flow to an Activity *Task3*
- Create a merge Parallel Gateway *Merge* and sequence flows from *Task1*, *Task2*, *Task3*, sequence flow to
- An EndEvent *EndEvent1*

Configure BPSim

In this section, we create the Artifact, Configure package and configure the parameter values of each element.

Artifact & Package	• Open BPSIm Manager • Create Artifact named "IntermediateEvent - Standalone- Timer" • Select Package of the model created Then all the BPMN elements will be loaded in to the manager.
StartEvent1	• From the tree in the BPSim Manager, navigate to StartEvent | StartEvent1 • Activate *Control* Tab • Expand the drop menu on "New Parameter..." and select "TriggerCount" • Give a value of 1 • Activate *Properties* Tab • Create a property "dummyProperty" with value 0; with this property, the property trace dialog will be able to show the sequence of element flows during

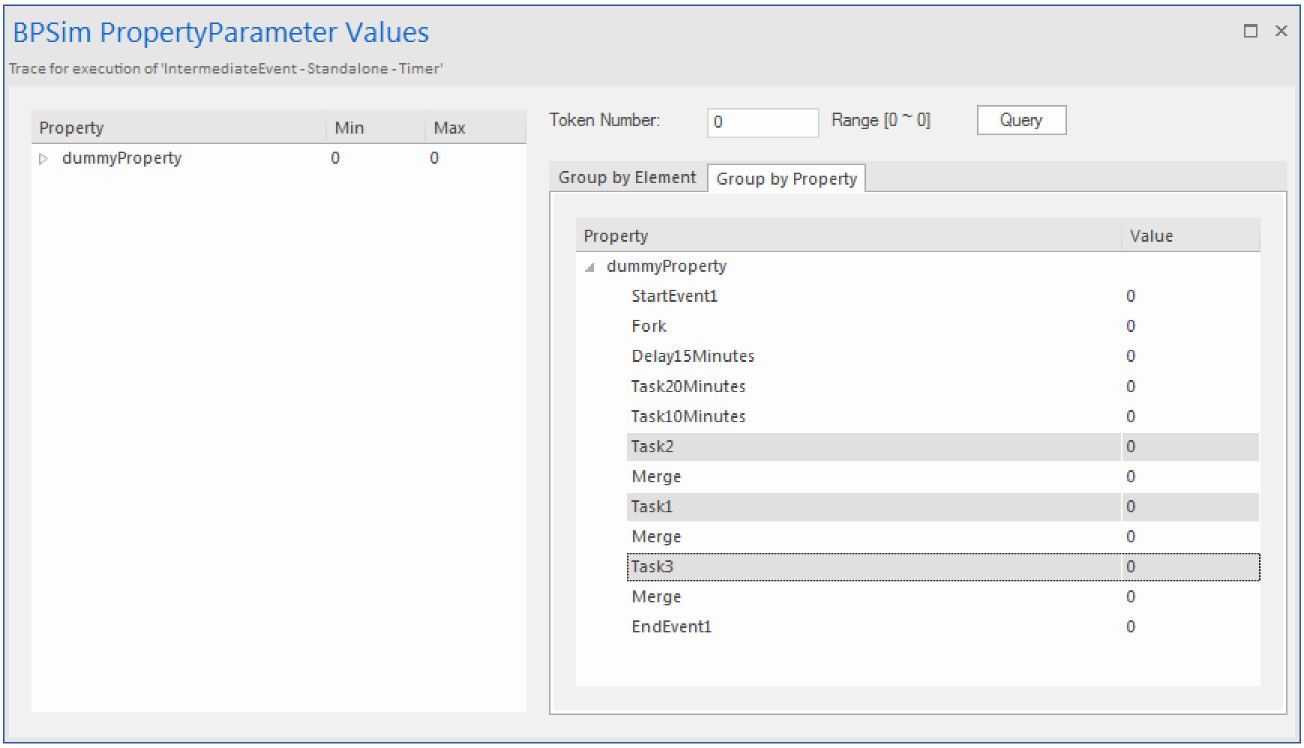
	simulation.
Delay15Minutes	- From the tree in the BPSim Manager, navigate to IntermediateEvent| Delay15Minutes - Activate *Control*Tab - Expand the drop menu on "New Parameter..." and select "InterTriggerTimer" Give the value 15 minutes, which shows as: 000:000:000 000:15:00
Task10Minutes	- From the tree in the BPSim Manager, navigate to Activity | Task10Minutes - Activate *Time* Tab - Expand the drop menu on "New Parameter..." and select "ProcessingTime" - Give the value 10 minutes, which shows as: 000:000:000 000:10:00
Task20Minutes	- From the tree in the BPSim Manager, navigate to Activity | Task20Minutes - Activate *Time* Tab - Expand the drop menu on "New Parameter..." and select "ProcessingTime" - Give the value 20 minutes, which shows as: 000:000:000 000:20:00

Run Simulation

Run Simulation

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog

- Run button | Standard Simulation
- After simulation, click the 3rd button on the toolbar to show the *BPSim PropertyParameter Values* dialog,
- click the *Query* button, activate the *Group by Property* page


BPSim PropertyParameter Values
Trace for execution of 'IntermediateEvent - Standalone - Timer'
Token Number:
0
Range [0 ~ 0]
Query
Group by Element
Group by Property

Property	Min	Max
dummyProperty	0	0

Property	Value
dummyProperty	
StartEvent1	0
Fork	0
Delay15Minutes	0
Task20Minutes	0
Task10Minutes	0
Task2	0
Merge	0
Task1	0
Merge	0
Task3	0
Merge	0
EndEvent1	0

Analysis:

The Fork Parallel gateway will activate the outgoing sequence flows simultaneously (the order is undefined and not important). However, we would expect the order among Task1, Task2, Task3 be exactly:

- Task2,
- Task1,
- Task3

The order is determined by the bpsim parameters set on the activities(ProcessingTime) and the timer intermediate event(InterTriggerTimer).

The sequence shown in the BPSim PropertyParameter Values dialog, shows that Task2 comes ahead of Task1, which comes ahead of Task3.

Help Desk Phone Support Simulation - Resource Perspective

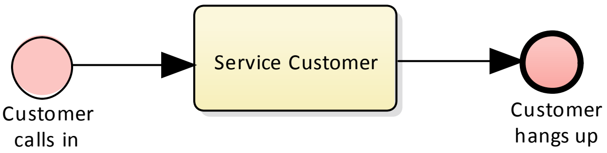
In this example, we create a very simple model to simulate the help desk phone support process.

We set up a circumstance that resource are limited and the requests has to be put in a waiting queue for resource. Then we try to seek a balance point between customer's waiting time and the amount of resources by a what-if analysis.

Firstly, we step-by-step model this process and start with a simple parameter setting, which is capable to be calculated with pen & paper, then verify with EABPSim. After that, we do some what-if analysis that may help the manager to make a decision.

Create BPMN Model

The model itself is very simple, a Start Event, a Task and an End Event.



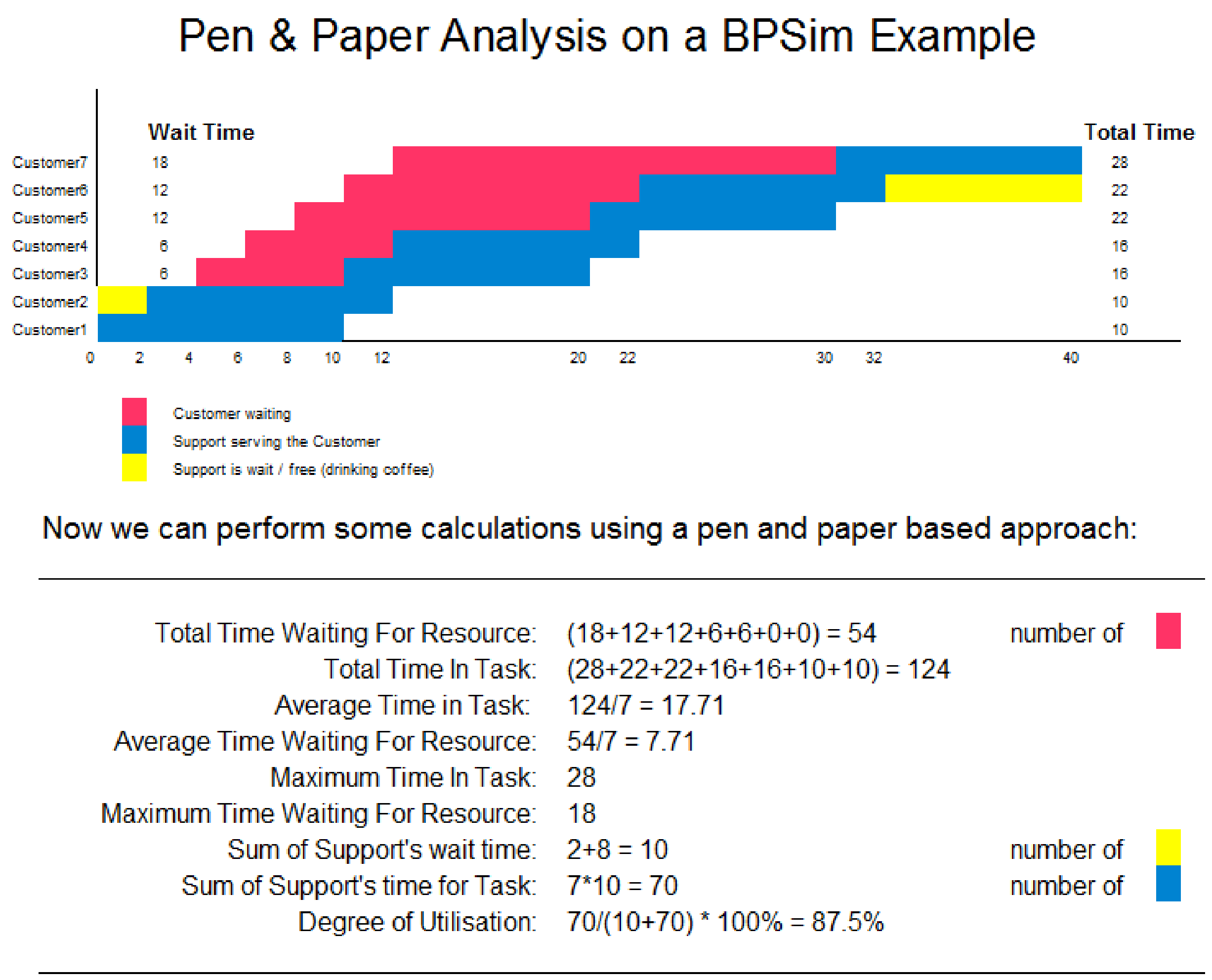
- Create a Start Event *Customer calls in*, sequence flow to
- an abstract task *Service Customer*, sequence flow to
- an End Event *Customer hangs up*

Create a BPMN2.0 Resource named *Support,* this element will be used in BPSim Configuration

Pen & Paper Analysis

We will use pen & paper to analyze the following case:

- 7 customers will call in every 2 minutes
- only 2 supports are available
- each service will take 10 minutes



Pen & Paper Analysis on a BPSim Example

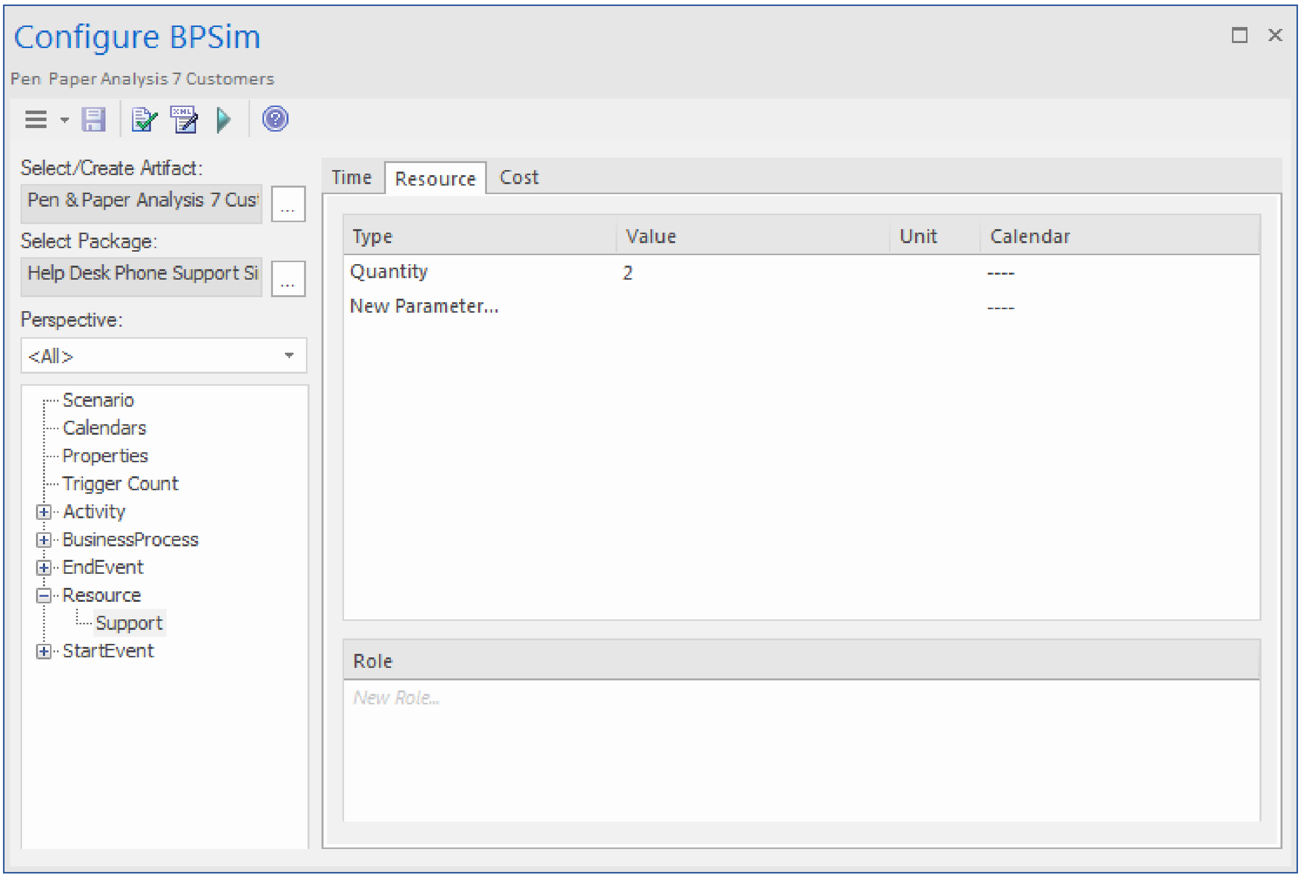
Now we can perform some calculations using a pen and paper based approach:

Total Time Waiting For Resource:	(18+12+12+6+6+0+0) = 54	number of 🟥
Total Time In Task:	(28+22+22+16+16+10+10) = 124	
Average Time in Task:	124/7 = 17.71	
Average Time Waiting For Resource:	54/7 = 7.71	
Maximum Time In Task:	28	
Maximum Time Waiting For Resource:	18	
Sum of Support's wait time:	2+8 = 10	number of 🟨
Sum of Support's time for Task:	7*10 = 70	number of 🟦
Degree of Utilisation:	70/(10+70) * 100% = 87.5%	

Seen from this result, it is already very complicated computation for such a simple model when resource constraints are applied. When the process expand and more constraints are applied, the ability of analyzing with pen & paper will quickly become inapplicable. We will show you how EABPSim can help.

BPSim Configuration

- Create a BPSim Configuration artifact *Pen & Paper Analysis 7 customers*, configure to the package containing this BPMN model
- Select Resource | *Support*, open Resource tab, create parameter Quantity with value 2

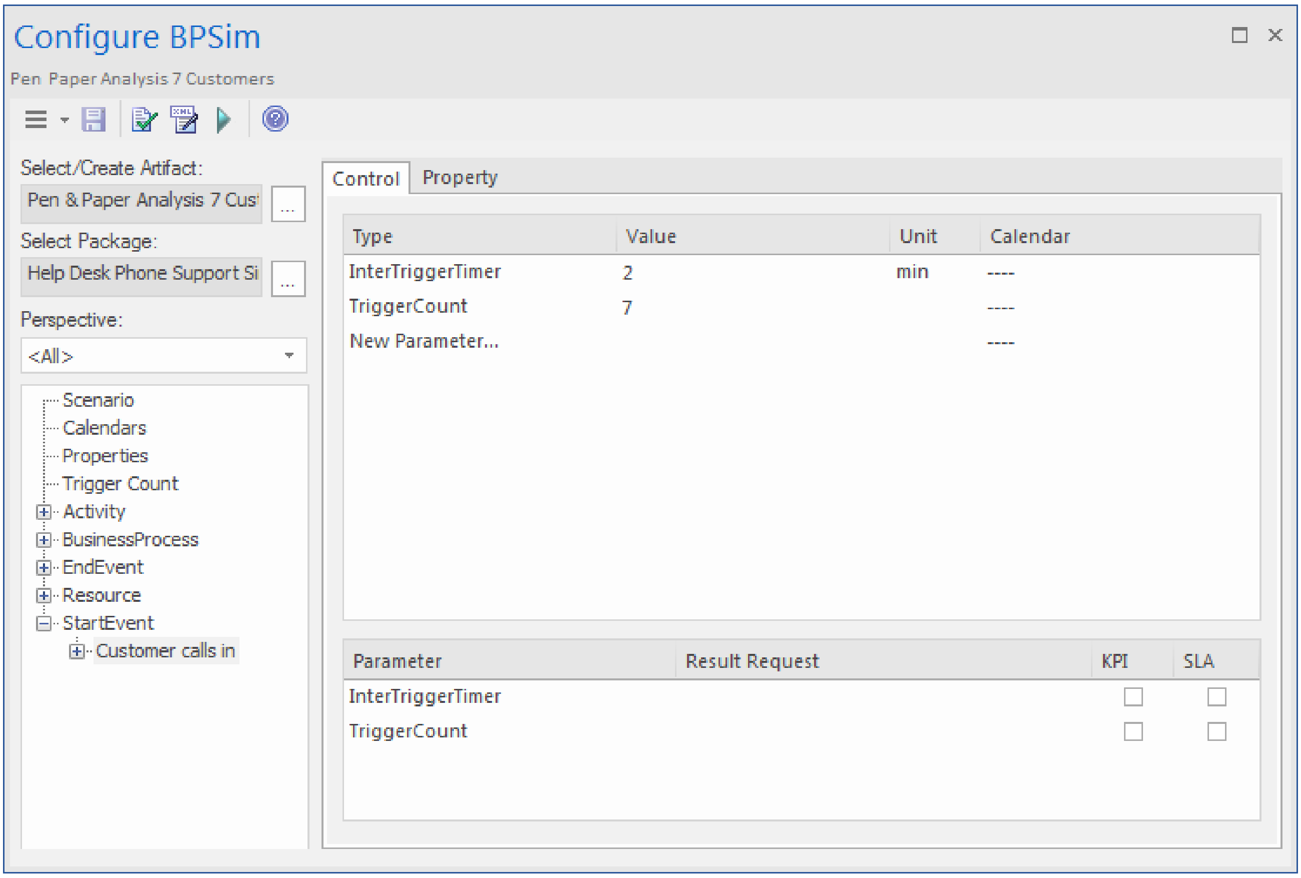


Type	Value	Unit	Calendar
Quantity	2		----
New Parameter...			----

- Select StartEvent | *Customer calls in*, open Control tab,

  - create parameter InterTriggerTimer with value 2 minutes

  - create parameter TriggerCount with value 7



Configure BPSim

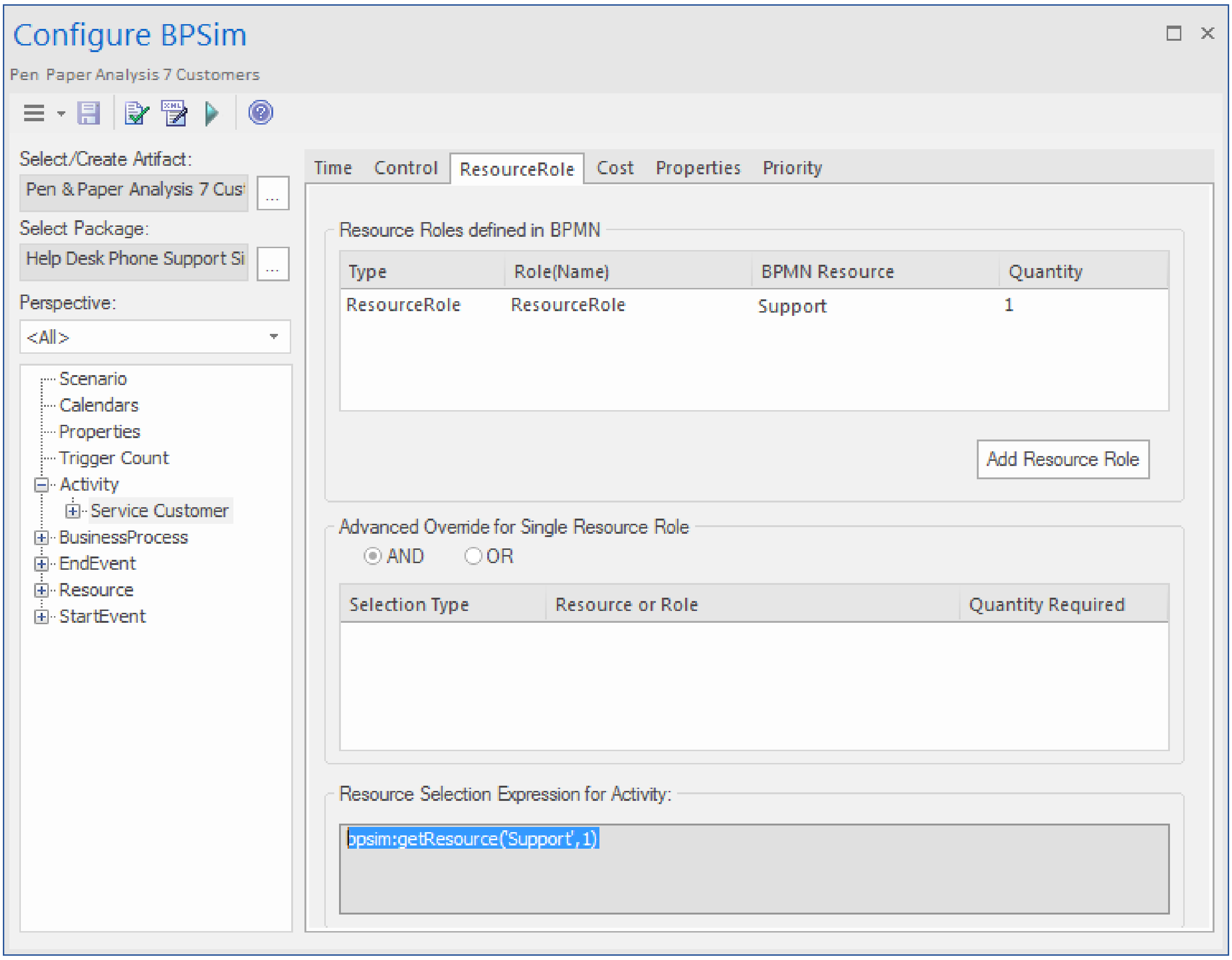
Pen Paper Analysis 7 Customers

Type	Value	Unit	Calendar
InterTriggerTimer	2	min	----
TriggerCount	7		----
New Parameter...			----

Parameter	Result Request	KPI	SLA
InterTriggerTimer		□	□
TriggerCount		□	□

- Select Activity | *Service Customer*, open Time tab, create parameter ProcessingTime with value 10 minutes
- Select Activity | *Service Customer*, open ResourceRole tab, click button "Add Resource Role", a new row will be created in list of "Resource Roles defined in BPMN" | click "..." button for BPMN Resource | select element *Support* we created earlier. Input value "1" in Quantity

In the group "Resource Selection Expression for Activity", you will see the automatically generated expression:



Resource Roles defined in BPMN

Type	Role(Name)	BPMN Resource	Quantity
ResourceRole	ResourceRole	Support	1

Advanced Override for Single Resource Role

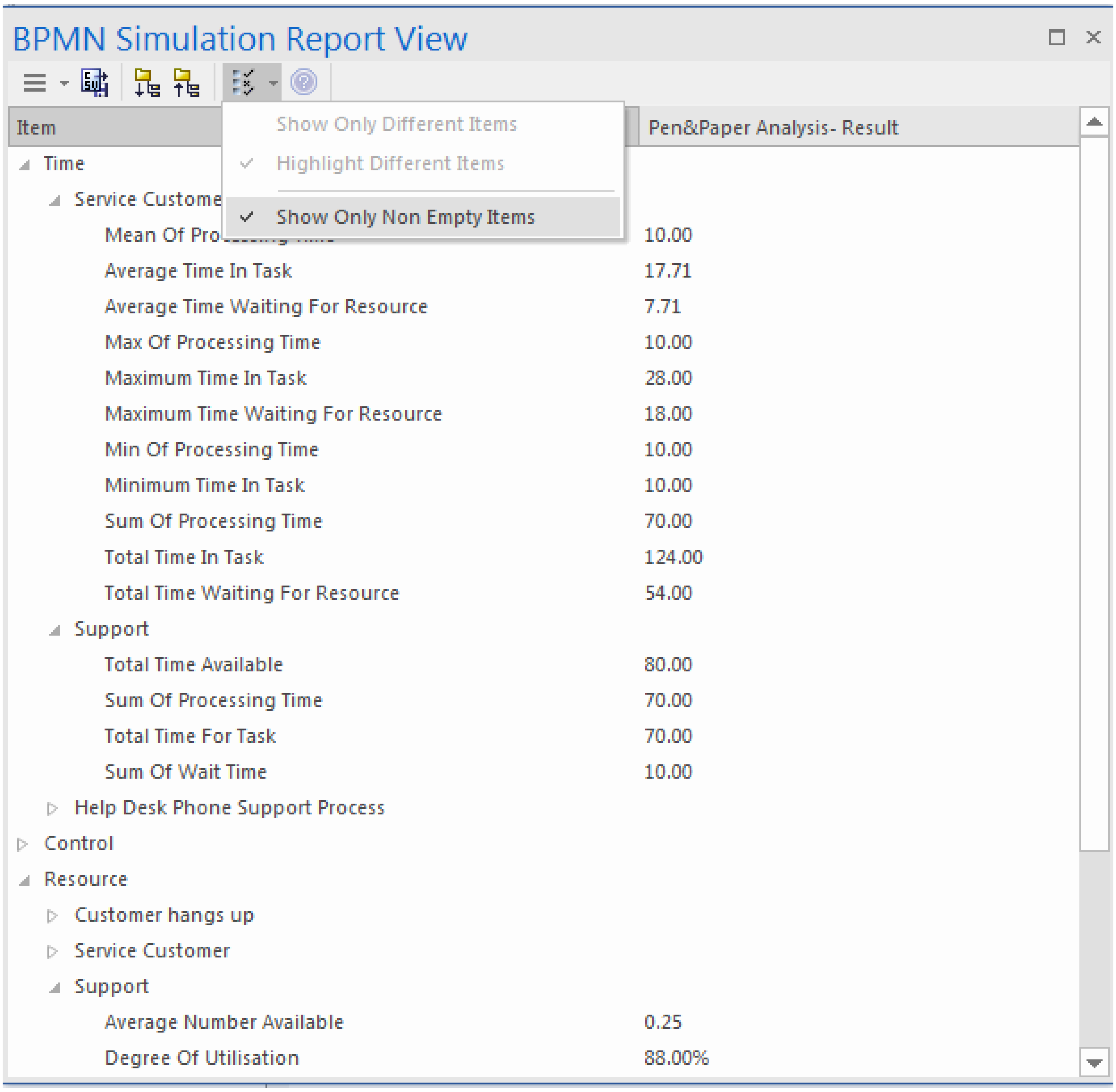
● AND ○ OR

Resource Selection Expression for Activity:

```
bpsim:getResource('Support',1)
```

Run Simulation and Analyze

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- On simulation, open the report element by clicking 4th button on the controller's toolbar


BPMN Simulation Report View
Show Only Different Items
✓ Highlight Different Items
✓ Show Only Non Empty Items

Item	Pen&Paper Analysis- Result
Time	
Service Custome	
Mean Of Pro	10.00
Average Time In Task	17.71
Average Time Waiting For Resource	7.71
Max Of Processing Time	10.00
Maximum Time In Task	28.00
Maximum Time Waiting For Resource	18.00
Min Of Processing Time	10.00
Minimum Time In Task	10.00
Sum Of Processing Time	70.00
Total Time In Task	124.00
Total Time Waiting For Resource	54.00
Support	
Total Time Available	80.00
Sum Of Processing Time	70.00
Total Time For Task	70.00
Sum Of Wait Time	10.00
Help Desk Phone Support Process	
Control	
Resource	
Customer hangs up	
Service Customer	
Support	
Average Number Available	0.25
Degree Of Utilisation	88.00%

Tips: you can use the "Show Only Non-Empty Items" to filter the results

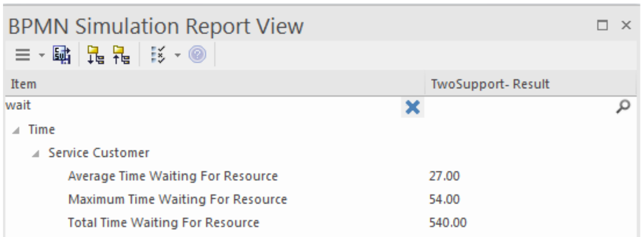
The result match our Pen & Paper Analysis.

Simulation - 2 supports for 20 customers

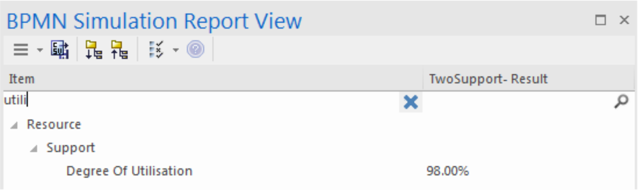
You can create an artifact by copy from an existing BPSim Configuration. Copy the *Pen & Paper Analysis 7 customers* and **Ctrl** + **Shift** + V to paste, with new name *TwoSupport*.

- Double click *TwoSupport* to open configuration manager, we can see that all the configurations are kept from the copy source.
- We browse to Select StartEvent | *Customer calls in*, open Control tab, change parameter TriggerCount with value 20.

Run simulation and Analyze


BPMN Simulation Report View

Item	TwoSupport- Result
wait	
Time	
Service Customer	
Average Time Waiting For Resource	27.00
Maximum Time Waiting For Resource	54.00
Total Time Waiting For Resource	540.00


BPMN Simulation Report View

Item	TwoSupport- Result
utili	
Resource	
Support	
Degree Of Utilisation	98.00%

Seen from the report, we can find that:

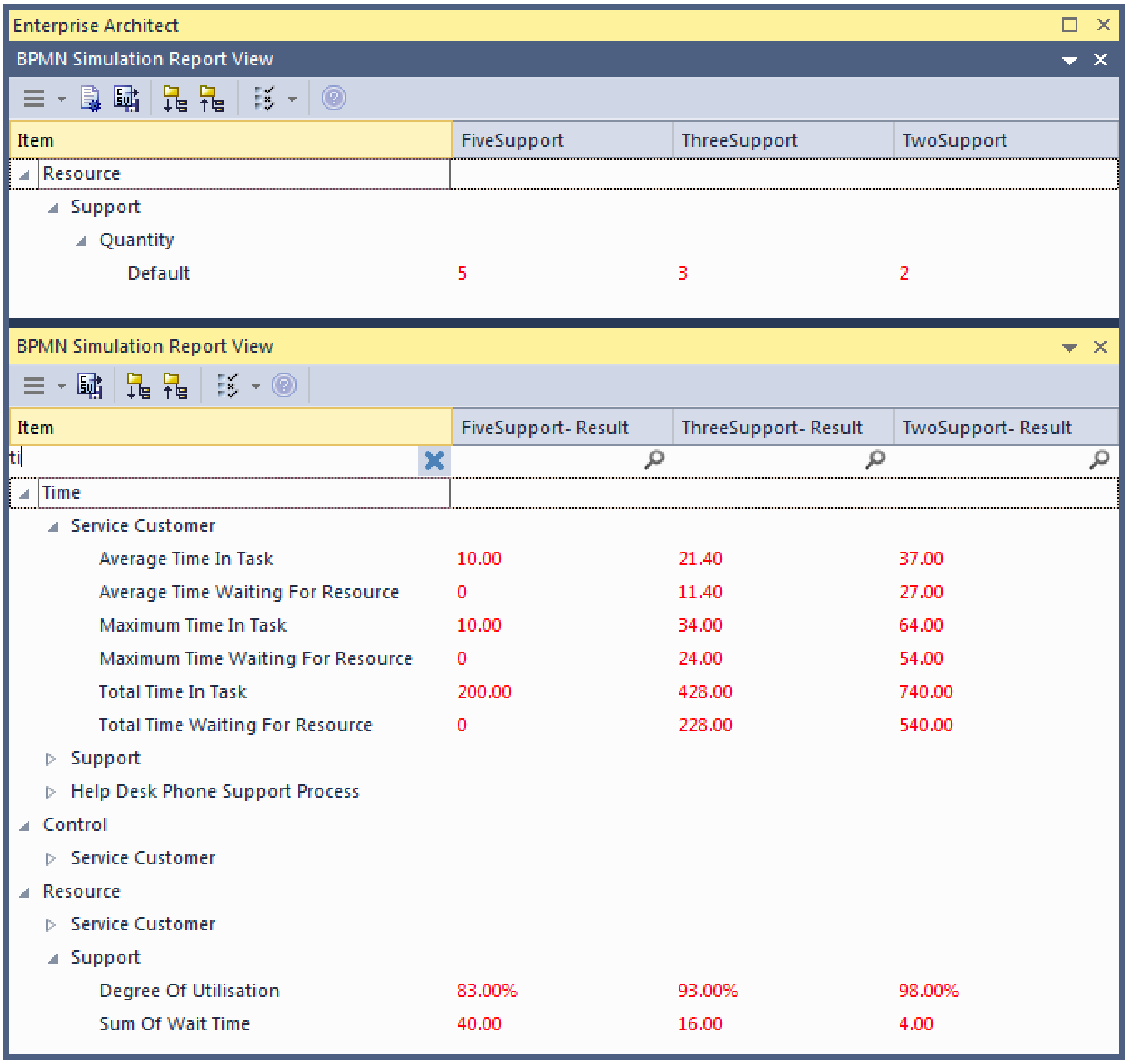
- the "Average Time Waiting For Resource" is 27 minutes and the "Maximum Time Waiting For Resource" is 54 minutes.
- How about the 2 supports, are they busy? If no, we might need to change the process to keep them busy to reduce the customer's waiting time. However, the "Degree Of Utilisation" already reached 98%. This indicated that the 2 supports almost had no idle time.

What-If I have more staff? Compare with 2, 3, 5 supports

- Copy *TwoSupport* and **Ctrl** + **Shift** + V to paste, with new name *ThreeSupport.*
- Copy *TwoSupport* and Ctrl + Shift + V to paste, with new name *FiveSupport.*
- Double click *ThreeSupport* to open the manager | browse to Resource | Support | Resource tab | change value of parameter Quantity from 2 to 3
- Double click *FiveSupport* to open the manager | browse to Resource | Support | Resource tab | change value of parameter Quantity from 2 to 5
- Run the simulation and do compare

  - select *TwoSupport*, *ThreeSupport* and *FiveSupport*, context menu | Show BPSim Configuration

  - select *TwoSupport- Result*, *ThreeSupport- Result* and *FiveSupport- Result*, context menu | Show BPSim Report



Item	FiveSupport	ThreeSupport	TwoSupport
Resource			
Support			
Quantity			
Default	5	3	2

Item	FiveSupport- Result	ThreeSupport- Result	TwoSupport- Result
ti			
Time			
Service Customer			
Average Time In Task	10.00	21.40	37.00
Average Time Waiting For Resource	0	11.40	27.00
Maximum Time In Task	10.00	34.00	64.00
Maximum Time Waiting For Resource	0	24.00	54.00
Total Time In Task	200.00	428.00	740.00
Total Time Waiting For Resource	0	228.00	540.00
Support			
Help Desk Phone Support Process			
Control			
Service Customer			
Resource			
Service Customer			
Support			
Degree Of Utilisation	83.00%	93.00%	98.00%
Sum Of Wait Time	40.00	16.00	4.00

Tips:

- *Use the 6th menu button and choose Show only different items for both view*
- *You can dock the views together, so they will give us a straight answer: THESE are the difference in results caused by THOSE configuration differences*
- *Toggle the filter bar to filter the items that you are interested.*

Analysis

The docked comparison views shows the configuration differences and the corresponding result differences.

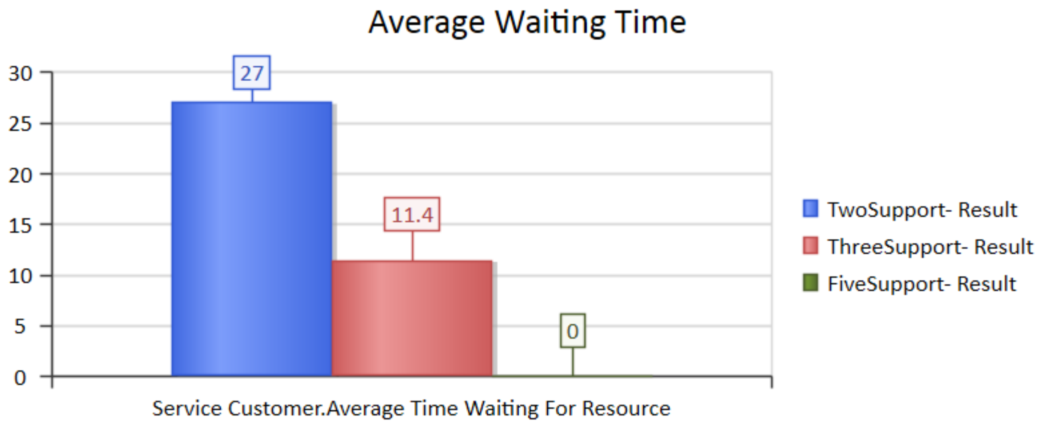
- The customer's waiting time dropped from 27 minutes(2 supports) to 11.4(3 supports) minutes and further down to 0 (5 supports)
- The "Degree Of Utilisation" dropped from 98% (2 supports) to 93%(3 supports) and further down to 83%(5 supports).

Of course the customers will be satisfied with 5 supports, however, the cost might be out of budget. So 4 supports might be a balance point for this case. You are recommended to copy an artifact and run a simulation for 4 supports.

Show Result with Chart

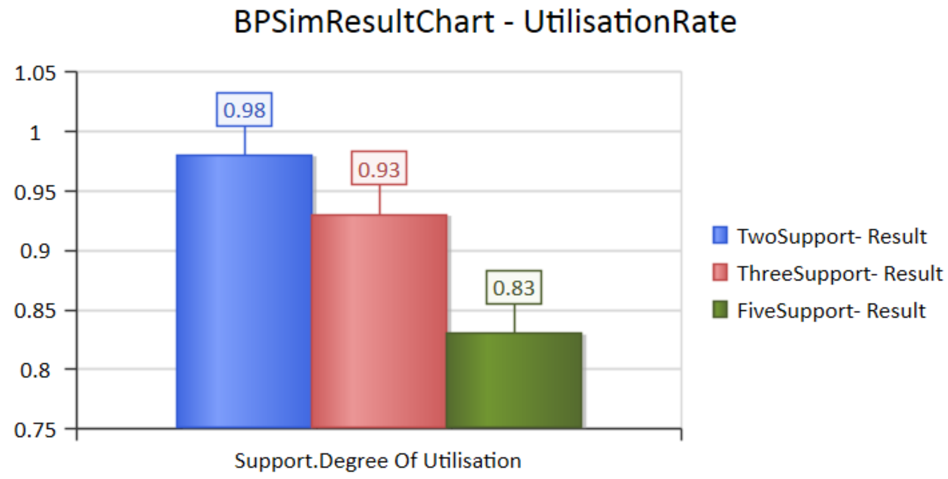
- Create a BPSim Result Chart on the diagram, name it *Average Waiting Time*
- Double click it to open the properties window | Open the BPSim Chart tab
- Click "..." button to select a Base Report, from which we define the schema (legends) we want to use in the chart. Select *TwoSupport-Result*
- Choose the following schema:

  - Time | Service Customer | Average Time Waiting For Resource
- Click "Add" to add another two BPSim Reports: *ThreeSupport- Result* and *FiveSupport- Result*
- OK, adjust the size of the chart element. This chart gave us very straight information.



- Create a BPSim Result Chart on the diagram, name it *Utilisation Rate*
- Double click it to open the properties window | Open the BPSim Chart tab
- Click "..." button to select a Base Report, from which we define the schema (legends) we want to use in the chart. Select *TwoSupport-Result*
- Choose the following schema:

  - Resource | Support | Degree Of Utilisation
- Click "Add" to add another two BPSim Reports: *ThreeSupport- Result* and *FiveSupport- Result*
- OK, adjust the size of the chart element. This chart gave us very straight information.


BPSimResultChart - UtilisationRate
1.05
1
0.95
0.9
0.85
0.8
0.75
0.98
0.93
0.83
Support.Degree Of Utilisation
TwoSupport- Result
ThreeSupport- Result
FiveSupport- Result

Paint Wall Process (simulate call activity)

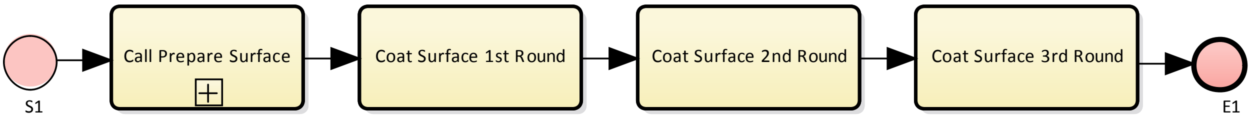
This is a simple process to simulate the process of painting a wall.

We define the main process by scaling the surface and then coating the surface 3 times.

Scaling the surface is further divided as multiple tasks like Scaling and cleaning.

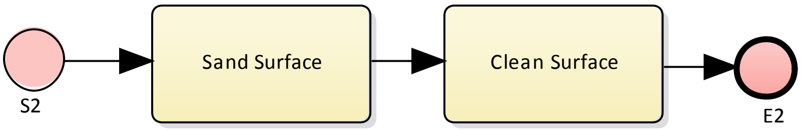
We also assume the three rounds of coating are the same, except that the time randomly spent on each task may be different.

Create BPMN Model



Create the main process - Paint Wall Process

- Create a Start Event *S1*, sequence flow to
- a callProcessActivity *Call Scale Surface*, sequence flow to
- a callGlobalTaskActivity *Coat Surface 1st Round*, sequence flows to
- a callGlobalTaskActivity *Coat Surface 2nd Round*, sequence flows to
- a callGlobalTaskActivity *Coat Surface 3rd Round*, sequence flows to
- an End Event *E1*



Create the re-used process - Scale Surface Process

- Create a Start Event *S2*, sequence flow to
- an abstract task *Scale Surface*, sequence flow to
- an abstract task *Clean Surface*, sequence flow to
- an End Event E2

Set Global Task and re-used process to call activities

- Create GlobalTask *Coat Surface*
- double click *Coat Surface 1st/2nd/3rd Round*, set the tag calledActivityRef to *Coat Surface*

Tips: you can also drag the Global Task *Coat Surface* from project browser and drop on the call activity element, click "set calledActivityRef" on the context menu.

- double click the *Call Scale Surface*, set the tag calledActivityRef to *Scale Surface Process*

Tips: you can also drag the process *Scale Surface Process* from project browser and drop on the call activity element, click "set calledActivityRef" on the context menu.

Configure BPSim

- Open BPSIm Manager
- Create Artifact named *Paint Wall Simulation*
- Select Package of the model created

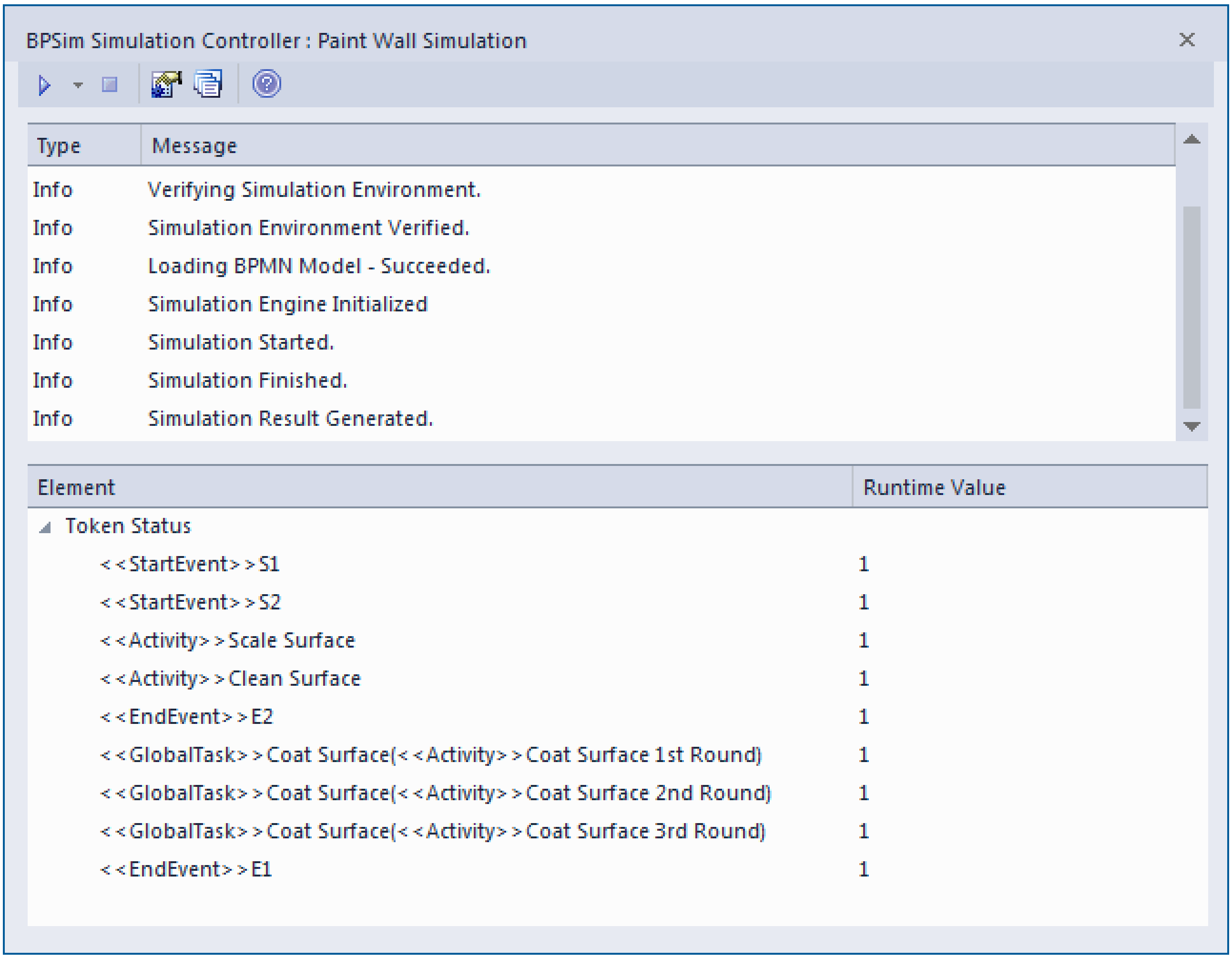
Then all the BPMN elements will be loaded in to the manager.

Fixed Scaling Time	• Select Activity *Coat Surface* | *Time* tab | Create Processing Time |input value 000:000:000 000:30:00, which means 30 minutes • Select Activity *Clean Surface* | *Time* tab | Create Processing Time |input value 000:000:000 000:10:00, which means 10 minutes
Random Coating Time	Select Global Task *Coat Surface* | Time tab | Create Processing Time | click "..." to open the parameter dialog | Distribution tab | Poisson | input "10" in the mean field | OK. With this setting, the mean value of the random numbers generated by this Poisson distribution is 10. Alternatively, You can choose other types of distributions.
TriggerCount on S1	Select Start Event *S1* | Control Tab | create TriggerCount | input value "1"

Run Simulation

Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | Standard Simulation
- On simulation, we will get the following result:



Type	Message
Info	Verifying Simulation Environment.
Info	Simulation Environment Verified.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>>S1	1
<<StartEvent>>S2	1
<<Activity>>Scale Surface	1
<<Activity>>Clean Surface	1
<<EndEvent>>E2	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 1st Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 2nd Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 3rd Round)	1
<<EndEvent>>E1	1

Flow Analysis

For the only token started on S1, we can see how it flows from the figure above:

- When reaching a callProcessActivity, the called process is activated. So we got *S2 ~ E2*
- When reaching a callGlobalTaskActivity, the called global task is activated. The notation reads: *GlobalTask name (called activity name)*. The global *Coat Surface* was called 3 times:

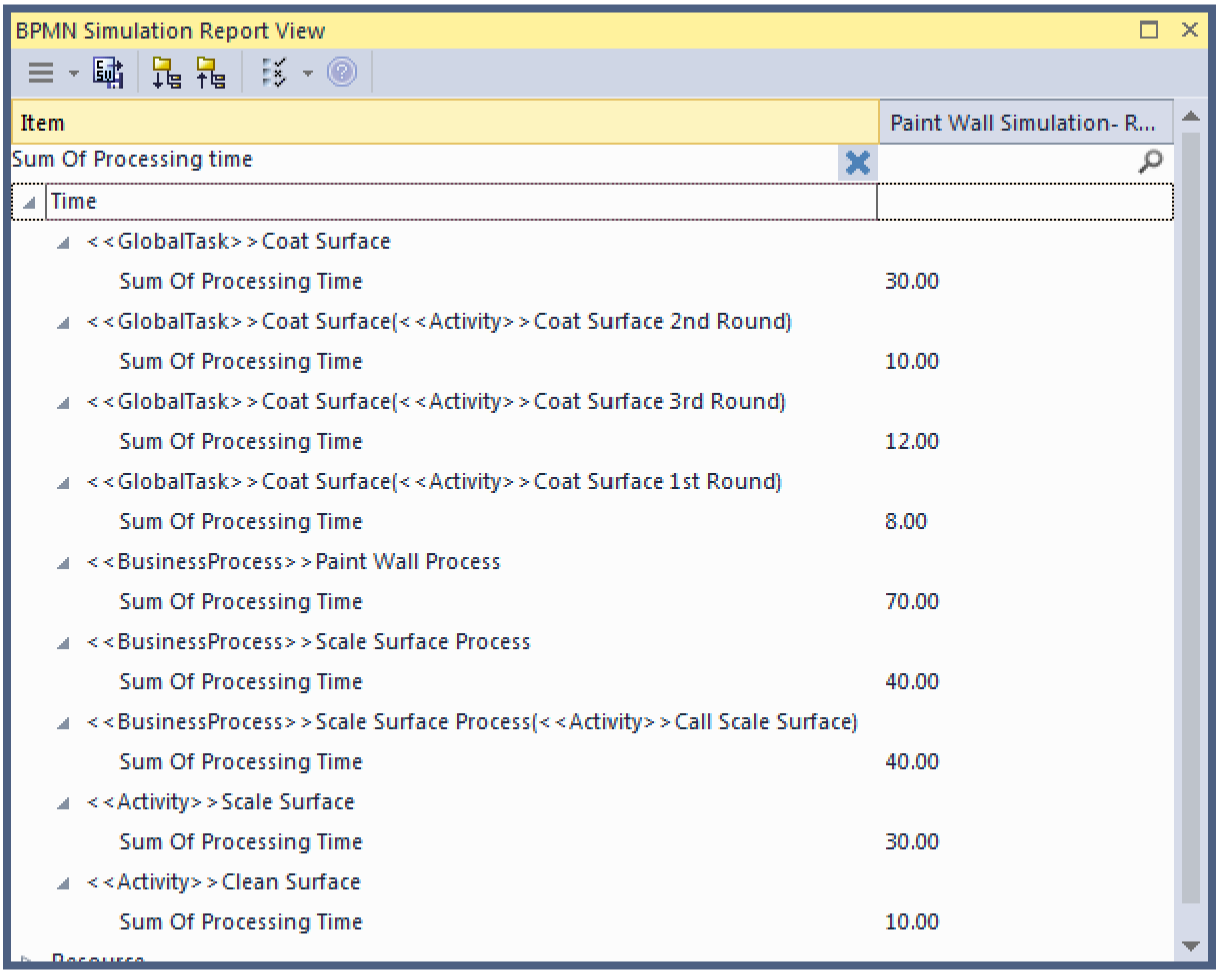
  - Coat Surface(Coat Surface 1st Round)

  - Coat Surface(Coat Surface 2nd Round)

  - Coat Surface(Coat Surface 3rd Round)

Time Analysis

Open the Report by click the 4th button on the tool bar:



Item	Paint Wall Simulation- R...
Sum Of Processing time	
Time	
<<GlobalTask>>Coat Surface	
Sum Of Processing Time	30.00
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 2nd Round)	
Sum Of Processing Time	10.00
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 3rd Round)	
Sum Of Processing Time	12.00
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 1st Round)	
Sum Of Processing Time	8.00
<<BusinessProcess>>Paint Wall Process	
Sum Of Processing Time	70.00
<<BusinessProcess>>Scale Surface Process	
Sum Of Processing Time	40.00
<<BusinessProcess>>Scale Surface Process(<<Activity>>Call Scale Surface)	
Sum Of Processing Time	40.00
<<Activity>>Scale Surface	
Sum Of Processing Time	30.00
<<Activity>>Clean Surface	
Sum Of Processing Time	10.00
Resource	

Tips: type in the filter bar will show only the matching records.

After expand (3rd button on the toolbar), we have result as the above figure, from which we can get the following analysis:

- *Coat Surface 1st Round* took 8 minutes; *Coat Surface 2nd Round* took 10 minutes; *Coat Surface 3rd Round* took 12 minutes; The figure 8, 10, 12 are randomly generated by the Poisson(10) distribution. What's important here is that: **each call instance of the global task has its own values**.
- *Coat Surface* has a total time collected from all instances: 8 + 10 + 12 = 30
- the call activity *Call Scale Surface* took 40 minutes, composed of *Scale Surface* (30 minutes) and *Clean Surface* (10 minutes)
- the Sum Of Processing Time for *Paint Wall Process* is 70 minutes, composed of the 4 call activities: 40 + 8 + 10 + 12 = 70

Customized Simulation

We can configure "Result Request" on BPMN Elements to customize the report, so we will only get the parameters we are interested in.

Configure Result Request:

- Select Activity | *Coat Surface 1st Round* | *Time* Tab | in the lower list select ProcessingTime | expand Request Request | check *Sum*
- Do the above steps on Call Scale Surface, Coat Surface 2nd Round, Coat Surface 3rd Round
- Do the above steps on BusinessProcess | Paint Wall Process

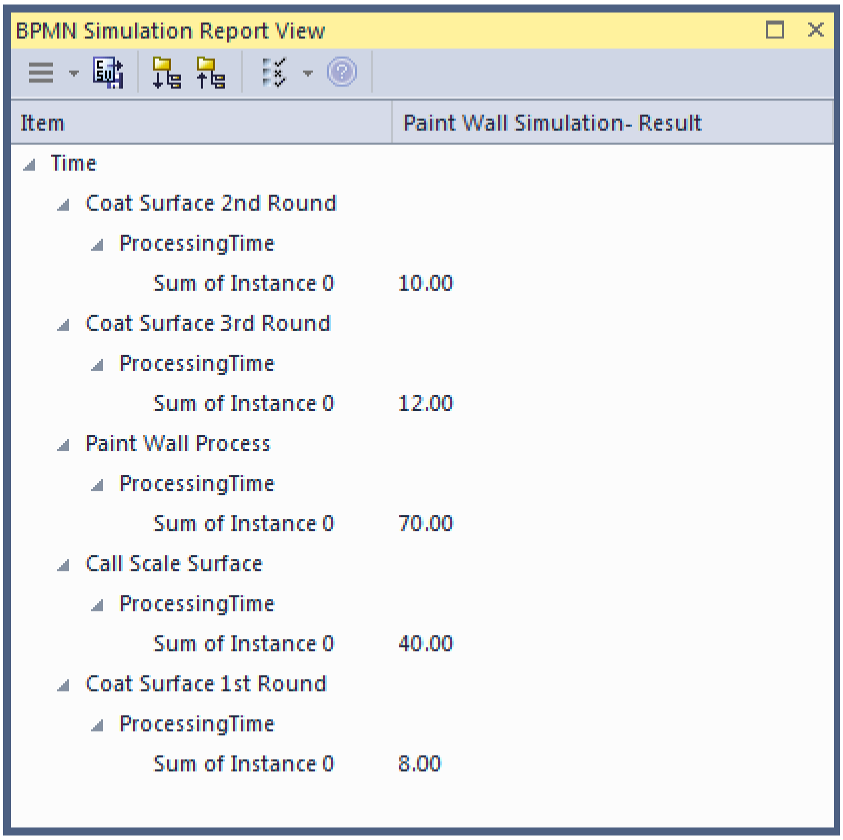
Run Simulation:

- From the Toolbar, click the **Run button** to open the BPSim Controller dialog
- Run button | **Customized Simulation**

The Flow Analysis is exactly the same as Standard Simulation.

Time Analysis

Open the Report by click the 4th button on the tool bar:



BPMN Simulation Report View

Item	Paint Wall Simulation- Result
Time	
Coat Surface 2nd Round	
ProcessingTime	
Sum of Instance 0	10.00
Coat Surface 3rd Round	
ProcessingTime	
Sum of Instance 0	12.00
Paint Wall Process	
ProcessingTime	
Sum of Instance 0	70.00
Call Scale Surface	
ProcessingTime	
Sum of Instance 0	40.00
Coat Surface 1st Round	
ProcessingTime	
Sum of Instance 0	8.00

The Time Analysis is exactly the same as Standard Simulation, however, the report only contains the result we requested.

Note: At the moment, we can not request ProcessingTime on the called process itself or the activities contained by the called process. If you have this requirement, use the Standard Simulation.

Export a BPSim Configuration

When you have defined a BPSim Configuration in a model, you can export it to an XMI file to be imported into other projects. The BPMN 2.0 model on which the configuration is based is also exported with the configuration. This binds the model to the appropriate BPSim Configuration when you import the XMI file into another project.

Access

Ribbon	Publish > Model Exchange > Export XMI > Export XMI : Publish
Menu	Package | Import/Export | Export Package to XMI : Publish
Context Menu	On Diagram or in **Project Browser** | Right-click BPSim Artifact | Export BPSim Configuration
Keyboard Shortcuts	**Ctrl** + **Alt** + E : Publish
Other	Toolbar of 'Configure BPSim' window | Export icon

Publish Model Package

The process of exporting a BPSim configuration and its model uses the '**Publish Model** Package' dialog for publishing a model to an XMI file.

Field/Option/Button	Description
Package	Defaults to the name of the Package containing the BPSim Artifact element.
Filename	Type in or browse for (click on the ... icon) the file path and XML filename into which to export the model.
XML Type	Select 'BPMN 2.0 XML'.
Export	Click on this button to export the configuration and BPMN 2.0 model. The export is complete when a confirmation message displays in the 'Progress' field.
Format XML Output	Defaults to selected; leave selected.
View XML	If you want to examine the exported XML, click on this button.

Notes

- To import the model from XMI into a new project, select the target Package in the new project and select 'Package | Import/Export | Import Package from XMI'