

## Data Modeling

<p>✓ Use natural English-sounding descriptions:</p> <ul style="list-style-type: none"> <li>• 'Process' is complete</li> <li>• 'Part' is not valid for 'Order' in 'Process Area'</li> <li>• The 'name' of person</li> </ul>	<p>✗ Do not use unnatural English-sounding descriptions:</p> <ul style="list-style-type: none"> <li>• Complete of 'Process'</li> <li>• Not valid 'Part' for 'Order' in 'Process Area'</li> </ul>
<p>✓ Use Title Case when naming objects</p> <ul style="list-style-type: none"> <li>&gt; Assertion Area</li> <li>&gt; Assertion Point</li> <li>Building</li> </ul>	<p>✗ Do not use plurals when naming object types</p> <p>Create a new object type called</p> <input type="text" value="Customers"/>
<p>✓ Always use lower case letters for simple property names</p> <p>PROPERTIES OF CUSTOMER &lt;Create new property&gt; delete pending flag ignore tag locations flag</p>	<p>✗ Do not use PascalCase or camelCase for names that are more than one word long</p> <p>Create a new object type called</p> <input type="text" value="AssertionArea"/> <input type="text" value="assertionArea"/>
<p>✓ Use spaces in names that are more than one word long</p> <p>&gt; Location Quality Monitor</p> <p>ignore tag locations flag remove location pending flag remove tag pending flag</p>	<p>✗ Do not use Hungarian Notation in property names</p> <p>Create a new property called</p> <input type="text" value="strRep2D"/> <input type="text" value="bDeletePendingFlag"/>
<p>✓ Use the naming convention (&lt;Type&gt; contains &lt;Type&gt;: bool) for complex properties that are booleans tracking the containment of another object</p> <p>'Process Area' contains 'Part' Bool</p>	
<p>✓ Use the enumeration type for a property that has a limited list of choices</p> <p>when the current area of <i>part</i> changes from <i>old</i> to <i>new</i> do if <i>new</i> is a <i>Storage Area</i> then set the status of <i>part</i> to</p> <p>Available Collected Delivered Error None</p>	

## Positioning Reps

✔ Put a small z-distance between 2D and 3D backgrounds to prevent rendering issues

Object  
3D Background.dae @ [0.00 0.00 -0.1]

Translation  
0  
0  
-0.1

✘ Do not include the file extension in the name when importing reps

Browse to a representation file  
C:\Users' \Downloads\PurpleStar.png Browse

Enter a name for this representation  
PurpleStar

Save Cancel

✔ Set the bottom value for stationary spaces to a number that is less than 0

Assertion Area 1

Choose the type of shape  
 Stationary  
 Moving

Top  
3

Bottom  
-0.1

Snap grid  
0.1

## Spatial Properties

✔ Set cell bottoms of object extents to at least -1

Forklift

Top  
1

Bottom  
-1

Snap grid  
0.1

Selected point  
0.000  
0.000

Save Cancel

✘ Do not set cell bottoms to 0 as they won't always appear in the location cell

Forklift

Top  
1

Bottom  
0

Snap grid  
0.1

Selected point  
0.000  
0.000

Save Cancel

✔ Use 'extent' when creating a container space, regardless if it is a moving or stationary object

✔ Use 'origin' when creating a contained space for a moving object

## Rules Creation

Add comments to Business Rules definitions

```
MHE contains Part ×
Defines the underlying logic that MHE contains Part becomes true when extent of MHE fully contains the origin of Part.
mhe contains part whenever
the extent of mhe contains the origin of part
```

Try to avoid using timeouts to create periodic events where possible

```
when the next update of clock is triggered
set the update triggered flag of clock to true ;
set the next update of clock to now + the update interval of clock
```

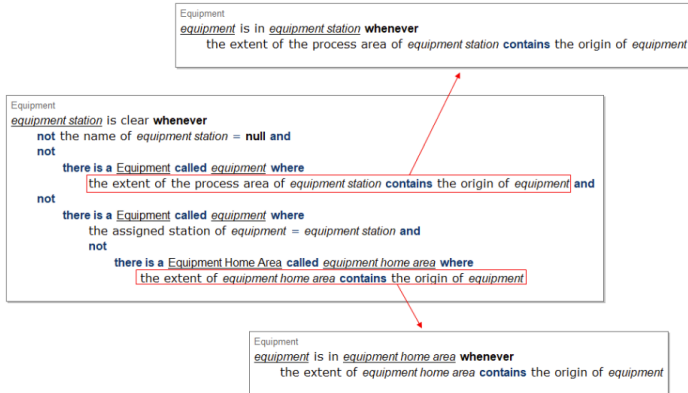
Add a description to Business Rules definitions before committing any changes

VERSION	STATUS	DESCRIPTION
Version 24	Superseded	track completion of order

If using timeouts to create periodic events, never calculate the next update time by adding an interval to the last update time

```
when the next update of clock is triggered
set the next update of clock to the next update of clock + the update interval of clock
```

Simplify complex rules by breaking them up into a series of smaller rules



```
when the next update of clock is triggered
set the next update of clock to now + the update interval of clock
```

Match the indexes of objects with its corresponding complex property

```
Equipment
equipment station is occupied whenever
there is a Equipment called equipment where
equipment station contains equipment
1 2
```

```
Equipment
equipment station is occupied whenever
there is a Equipment called equipment where
equipment is in equipment station
1 2
```