dunstan thomas

Working with BPMN 2.0 in Enterprise Architect version 11 / 12

Introduction

In versions 11 and 12 of Enterprise Architect several changes have been made to **BPMN 2.0** which affects the way that modellers use this notation and in particular how the repository must be structured.

An excellent video which summarises these changes is available at
http://www.sparxsystems.com/resources/webinar/partners/bpmn/bpmn-introduction-cephas.html

This article summarises these changes and presents a Tutorial so that you can have a hands-on experience of these changes and hence model effectively using BPMN 2.0 and Enterprise Architect versions 11 and 12.

Since **Collaboration** and **Process** diagrams are the diagrams most commonly used in Business Process Modelling, this article concentrates on these diagrams.

Summary of Changes

The changes made to BPMN 2.0 in Enterprise Architect version 11 and 12 are as follows:

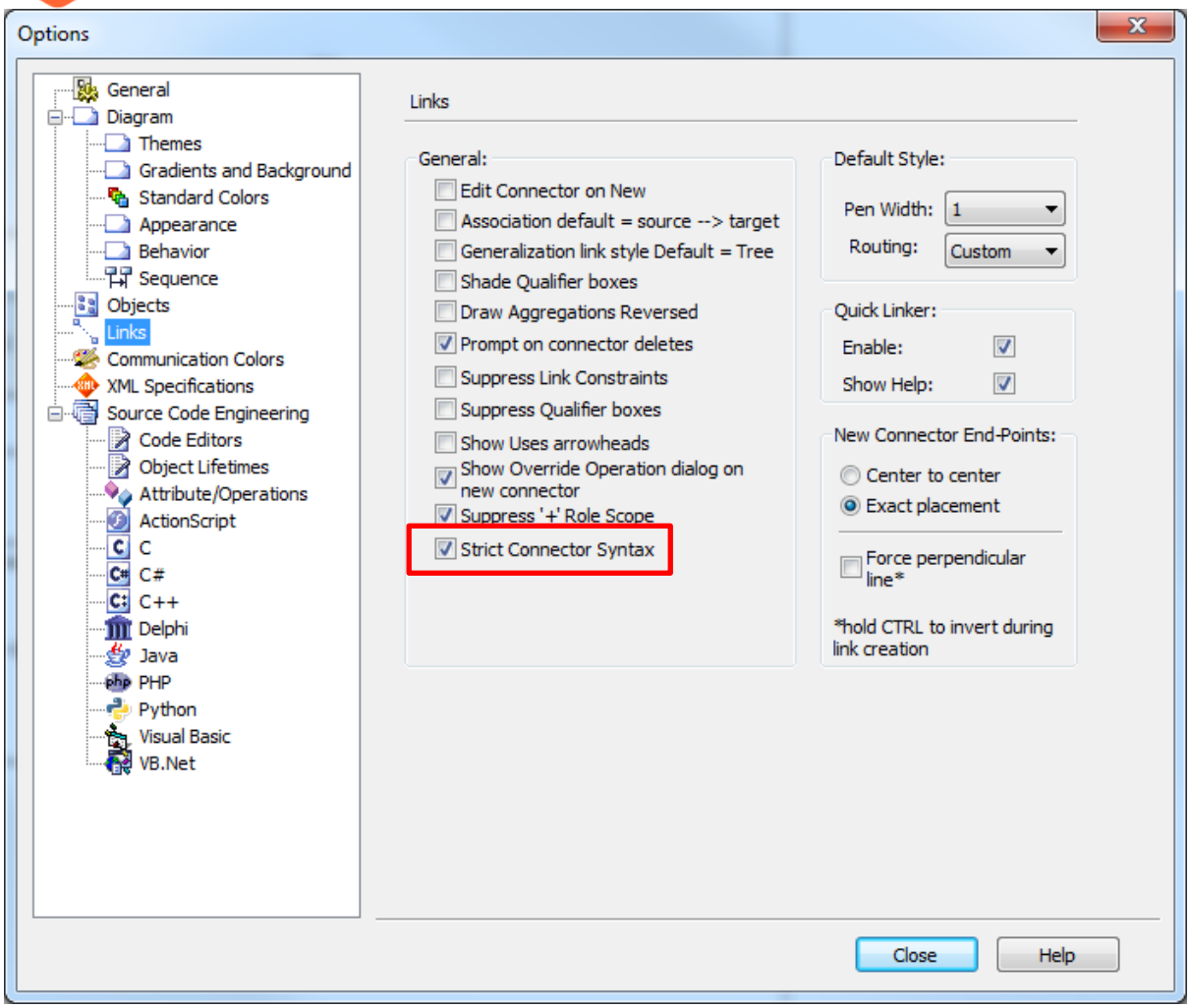
1) Closer compliance with the BPMN 2.0 specification (this affects re-usability of elements and repository structure)
2) Modelling interface improvements affecting:
    a. Diagrams (option to automatically re-size pools / lanes)
    b. Easier and more intuitive use of BPMN 2.0 tagged values in element dialogs
    c. Start and End events no longer produce a menu (**Stand alone | Edge Mounted**)

At first glance these may not seem significant, but they are, particularly change 1)

In order to gain maximum benefit from these changes it is **strongly** recommended that the modeller uses **Strict Connector Syntax** so that BPMN 2.0 conformant models are created.

Strict Connector Syntax is enabled by:

1) Selecting **Tools | Options…** from the main menu
2) Selecting **Links** and then checking the **checkbox** labelled **Strict Connector Syntax** as shown below



Compliance with BPMN 2.0 Specification

The main effect of this change is that **Activities remain as children of the enclosing Pool / Lane** and **CANNOT be placed in a separate package for re-usability**.

Pools and **Lanes**, may be placed in separate packages, but **are not re-used as Links**, rather they are **cloned** for use on multiple diagrams.

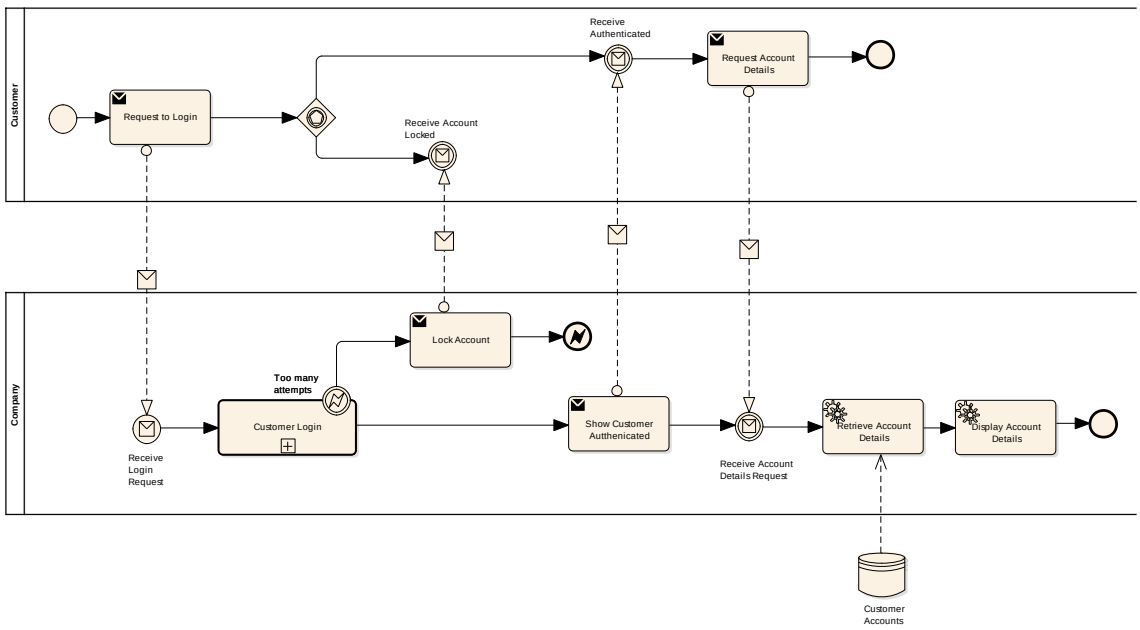
Global Activities (that is re-usable Activities) must be a **Sub-process** with a linked diagram which models the process. This structure should be placed in its own package. These **Global Activities** are then re-used by creating a **Call Activity** which refers to the **Global Activity**.

Worked Example ~ Tutorial

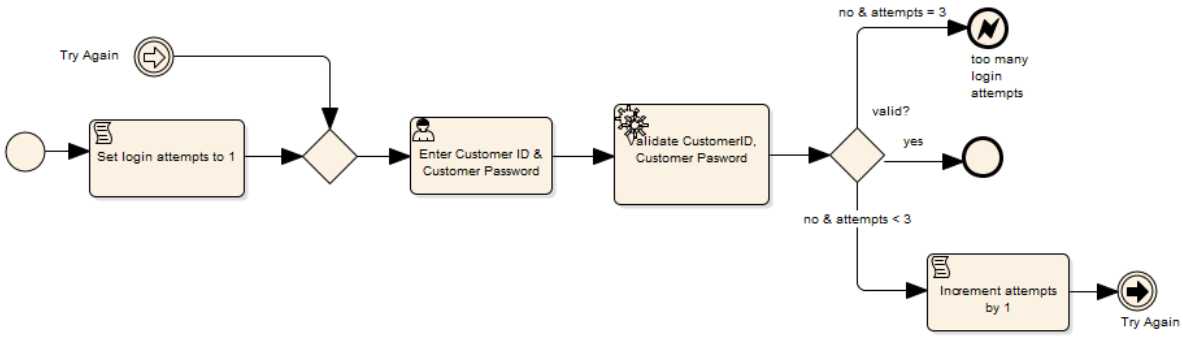
To illustrate how these changes affect the modeller I present a simple worked example which involves a Customer logging on to a company site and checking their account details.

The final Business Process flows are as shown below:

Request Account Details Process



Customer Login Process

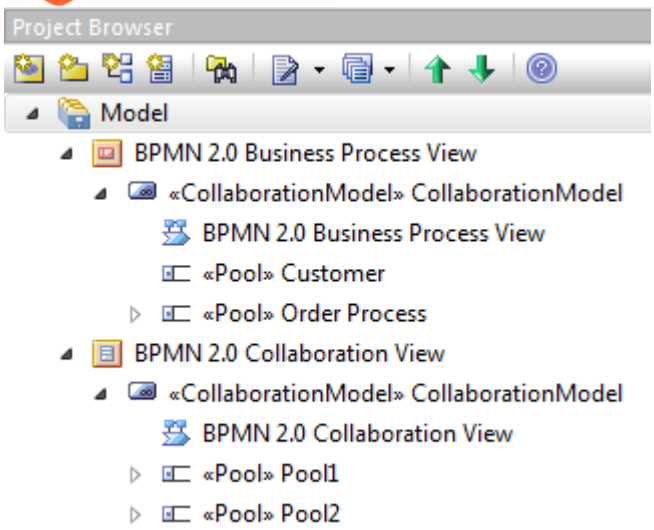


Step 1 – Create the Repository Structure

After creating a new repository (or Project as Enterprise Architect calls it), the **Model Wizard** for **BPMN 2.0** is used to create the initial structure. For this example I have used just the following sections from the **BPMN 2.0 Model Wizard**

- Collaboration
- Business Process

Resulting in the structure below:



Note how Enterprise Architect has created the structure and introduced two elements of type **<<CollaborationModel>> Collaboration Model**

Just as an aside there follows a brief discussion regarding this **CollaborationModel** and a similar element the **BusinessProcess**

*These elements **appear** to be an invention of Enterprise Architect. I say appear to be, since I cannot find an equivalent in the BPMN 2.0 specification, but I stand to be corrected on this matter.*

The usefulness of these elements seems to be a topic of some debate amongst Enterprise Architect users. I offer my own opinion here.

*Personally I see **no major advantage** in using them, since:*

- *perfectly stable and accurate BPMN 2.0 models can be constructed without them*
- *They add an extra hierarchy layer to the Repository Structure which may make (RTF) documentation generation more complex.*
- *However, each of these elements does contain **BPMN 2.0 Tagged Values** which allow the modeler to set **references** from a **CollaborationModel** to a **Choreography** and similarly set a **reference** from a **BusinessProcess** to a **Conversation**. This may be useful if you use **Choreography Diagrams** and/or **Conversation Diagrams**.*
- *It is possible (although I have not confirmed this), the use of these elements may be mandatory for **BPEL code generation**)*
- *Finally, in my opinion, the **CollaborationModel** created in the view **BPMN 2.0 Process View** should in fact be a **BusinessProcess** element*

I will let you decide whether or not to use these elements, for the purpose of this example, I will not be using them.

I prefer to begin modelling with empty views and packages and so I begin by deleting the **sample elements** created by the **Model Wizard**.

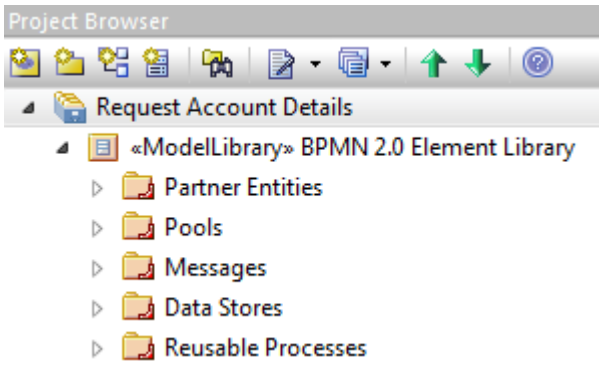
To use **BPMN 2.0** effectively I propose the use of a **Model Library** which will contain **packages** one for each BPMN 2.0 element that is deemed to be reusable. For this example I will need the following

- Partner Entities (who is involved – that is the name of the Pool)
- Pools (each Pool refers to a Partner Entity)
- Messages (these detail the data pertaining to a **Message Flow**)
- Data Stores
- Reusable Processes

Other packages worthy of consideration are

- Participants (used to allow **Pools** to refer to **Participant** elements on a **Conversation Diagram**)
- Resource Roles (departments or sections within a Participant)
- Lanes (each Lane refers to Resource Role)
- Data Objects (data that is produced or consumed during a Process)

After these changes have been made my Repository Structure now looks as below:



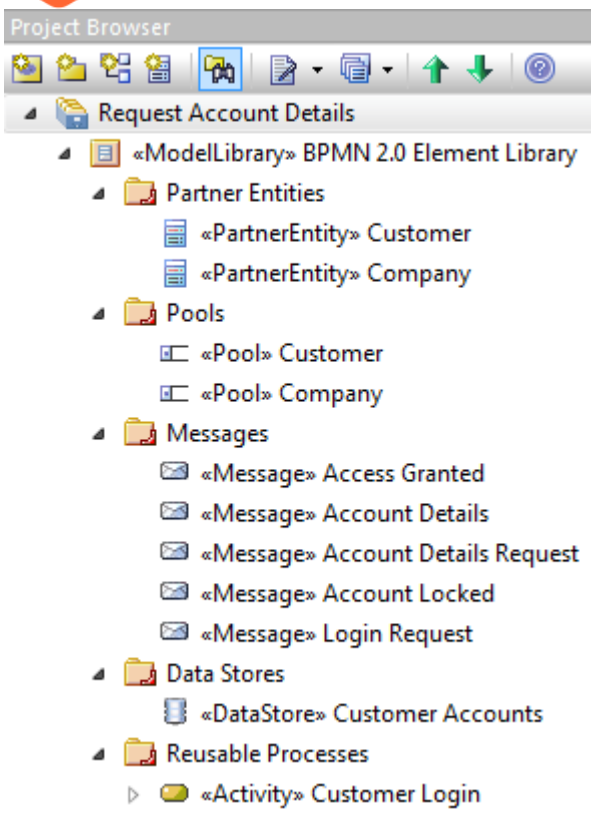
Note that I have set **Namespace Root** for all library packages, this will enable elements to be located more quickly when references are required to be set.

Step 2 – Populating the Library

Elements can be added to these library packages by either of the following methods:

1) Adding the elements directly into their respective package using **Add New Element.**
2) Create a diagram in the package and add elements using the diagram toolbox.

After populating the library my repository is as shown below:



For each **Pool** element in the **Library**:

- Set the **Pool**'s **BPMN 2.0 Tagged Value** named **partnerEntity** to the corresponding **Partner Entity** element in the library

If the name of the **Pool** needs to be changed, **change the name of the referenced Partner Entity** element. This has the effect of reflecting this name change to **all** reuses of the **Pool**

For each **Lane** element (no Lanes are used in this example)

- Set the **Lane**'s **BPMN 2.0 Tagged Value** named **partitionElementRef** to the corresponding **Resource Role** element in the library

If the name of the **Lane** needs to be changed, **change the name of the referenced Resource Role** element. This has the effect of reflecting this name change to **all** reuses of the **Lane**

Step 3 – Modelling the Reusable Process

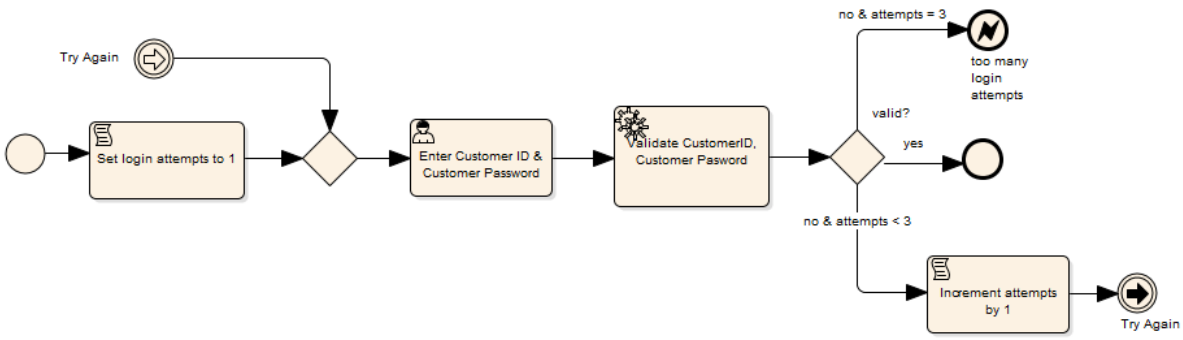
The first step is make the **Activity** named **Customer Login** which has just been added to the Element Library, a **Sub-Process**. This is achieved by:

1) Opening the properties dialog for the Activity element

2) Selecting **Sub process** from the **Type** dropdown list on the **BPMN 2.0 Tagged Values**. (Note how access to BPMN 2.0 Tagged Values has changed in EA version 11 /12)

The next step is to add a **BPMN 2.0 Business Process Diagram** (since there are no Pools) as a **direct child** to this Activity. (Simply select the Activity and **Add a New Diagram**)

Then model the following process flow as usual, noting the changes to the **BPMN 2.0 Tagged Values** for each element in the process flow.



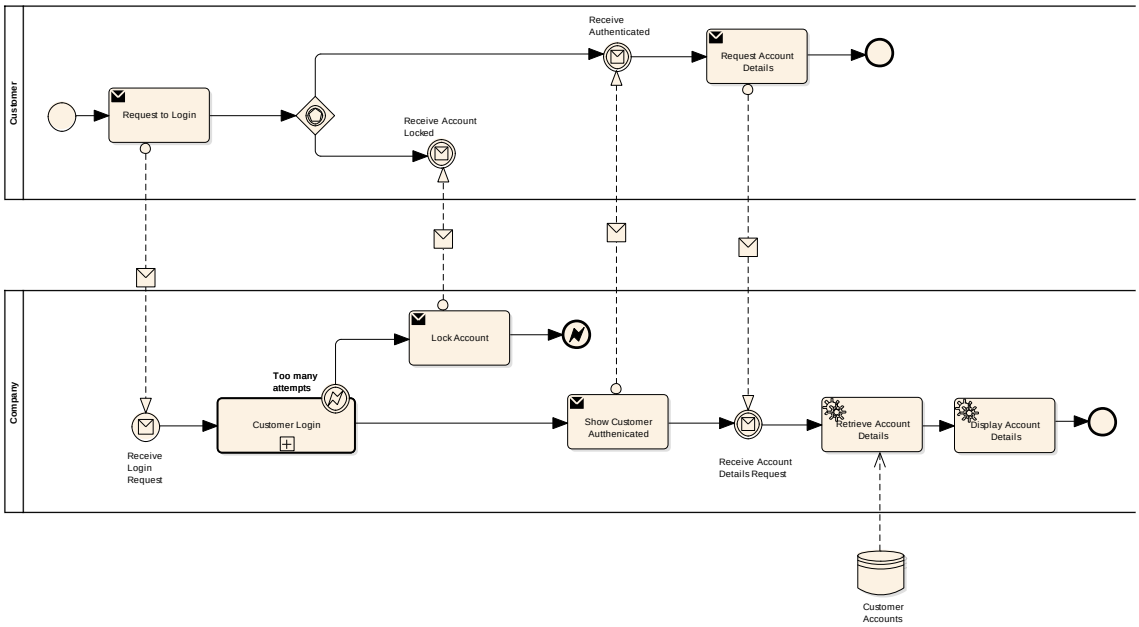
Finally, make the **Activity** named **Customer Login** a **Composite Element** by:

1) Adding the Activity to the diagram just created as a **Link**.
2) Right-click and select **New Child Diagram | Select Composite Diagram.**
3) Navigate to and select the diagram just created.
4) Delete the **Activity** named **Customer Login** from this diagram.

Step 4 – Modelling the Main Process Flow

The last step in this example is to create the main process flow.

1) Add a new **Package** named **Request Account Details** to the **View** named **BPMN 2.0 Collaboration View**.
2) Add a **BPMN 2.0 Collaboration Diagram** (since there are Pools) to this package.
3) Reuse the **Pools**, by
    a. **Copying the Pool element in the Library package to the clipboard and pasting into the package containing the diagram**.
    b. Deleting the name of the newly pasted **Pool** element (note that a name does appear on the diagram … this is taken from the **BPMN 2.0 Tagged Value** named **partnerElement**
4) (**Lanes** are also reused by using the **Copy and Paste** method)
5) Right-click the diagram and select **Set diagram flow direction | Horizontal**) (or **Vertical** if you prefer).
6) Add the Pools (from the diagram package) as **Links** on this diagram.
7) Model the process flow as shown below:



Notes

When modelling this process flow take note of the following:

1) As you add elements into their respective **Pool**, the **Pool** will re-size automatically to accommodate the new element.
2) Use the **messageRef BPMN 2.0 Tagged Value** to set the reference for the **Message Flow** to the appropriate **Message Element** in the library (it should be obvious in this example which Message elements are referred to)
3) When creating the **Called Activity** named **Customer Login** perform the following:
    a. Add a new **Activity** to the **Pool** named **Company**
    b. Name this Activity **Customer Login**
    c. Select **callActivity** from the **Type** dropdown list for the **BPMN 2.0 Tagged Values**
    d. Set **calledActivityRef** to the **Activity** named **Customer Login** located in the element library package named **Reusable Processes**
    e. Make this Activity a **Composite Element** and set its diagram to the **BPMN 2.0 Business Process** named **Customer Login** which is a child of the **Activity** named **Customer Login** located in the element library package named **Reusable Processes**

Conclusion

The simple example has demonstrated the changes made to BPMN 2.0 in Enterprise Architect versions 11 and 12. In particular:

- Changes to how elements such as **Pools** (and **Lanes**), and **Activities** are reused and how they reference **Partner Entity** and **Resource Role** elements respectively.
- Changes to the element property dialog regarding accessing **BPMN 2.0 Tagged Values**

I hope this example illustrates these changes clearly and assists you in creating well-structured and accurate BPMN 2.0 models in the future.

Phil Chudley
Principal Consultant
Dunstan Thomas Consulting