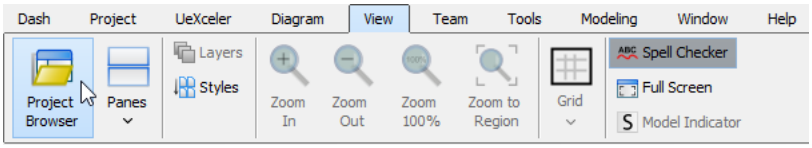
How to Organize Domain and Implementation Model?

Written Date : May 04, 2016

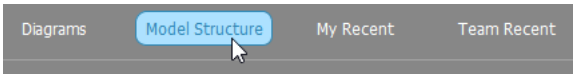
Creating a Domain Class Model

1. Create a new project by selecting **Project > New** from the application toolbar. In the **New Project** window, enter *Order Processing* as the project name and click **Create Blank Project**.

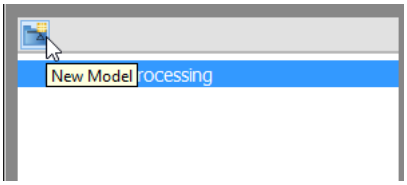
2. Open the **Project Browser** by selecting **View > Project Browser** from the application toolbar.



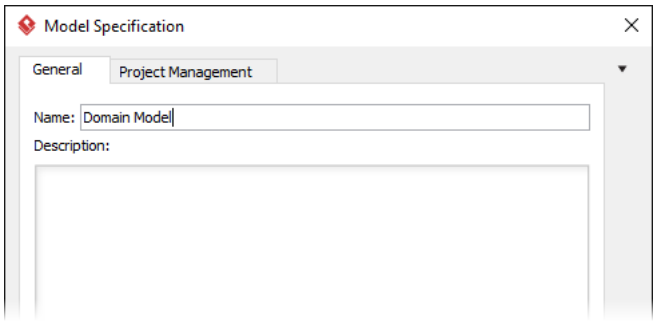
3. Open the **Model Structure** page.



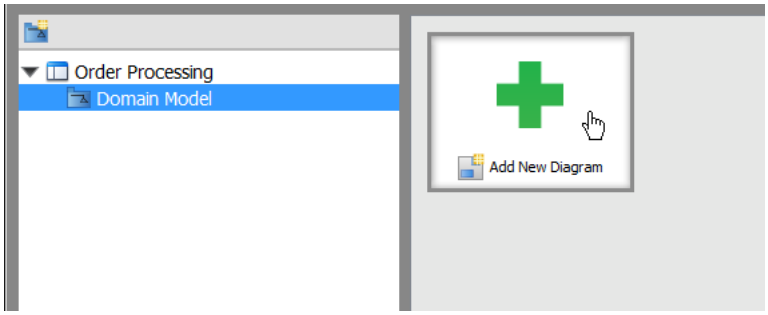
4. On the left-hand side, keep the project node selected in the list and then click **New Model**.



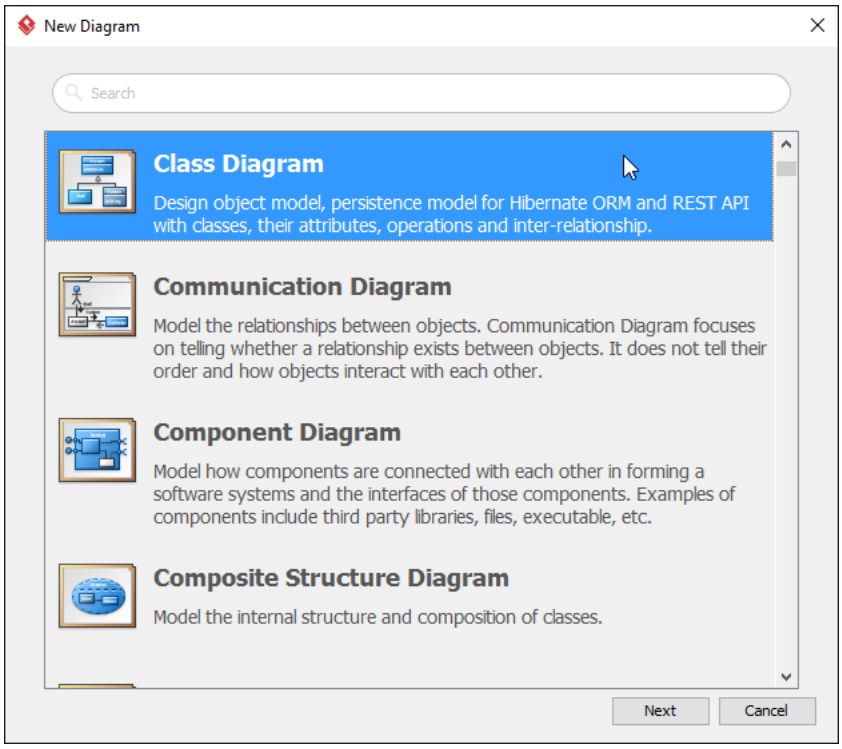
5. In the **Model Specification** window, enter *Domain Model* as the model name. Click **OK** to confirm.



6. Keep **Domain Model** selected in the list. Then, click **Add New Diagram**.

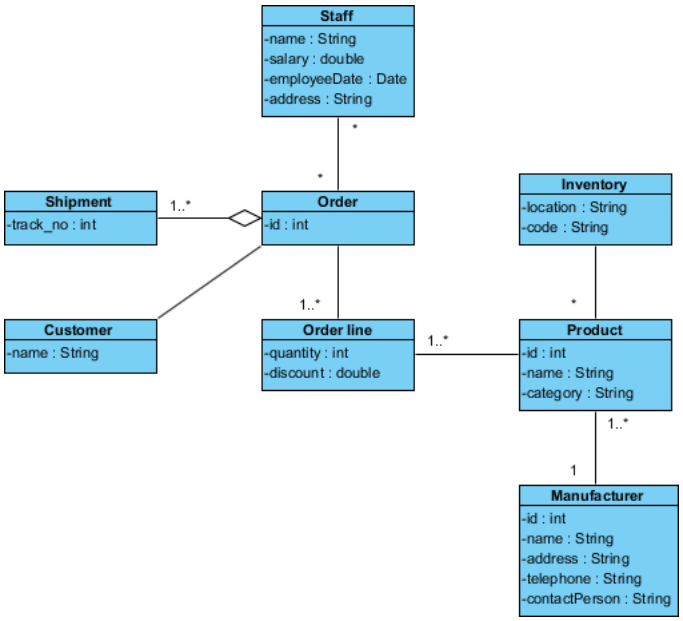


7. In the **New Diagram** window, select **Class Diagram** and then click **Next** to continue.



8. Keep the diagram name unchanged and then click **OK** to create the diagram.

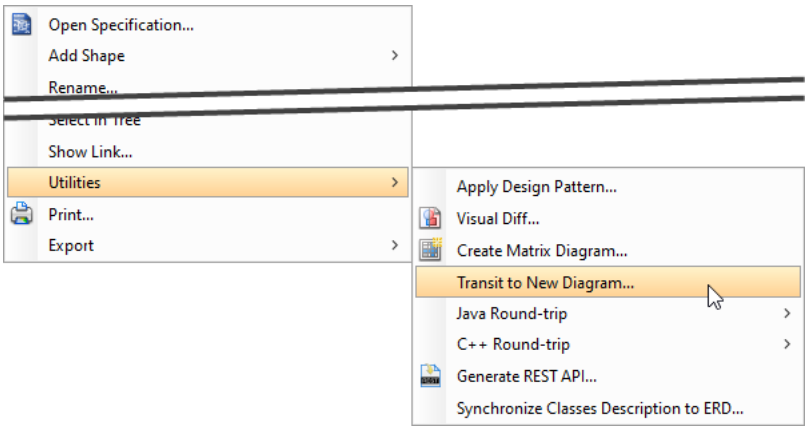
9. Draw a domain model class diagram like this:



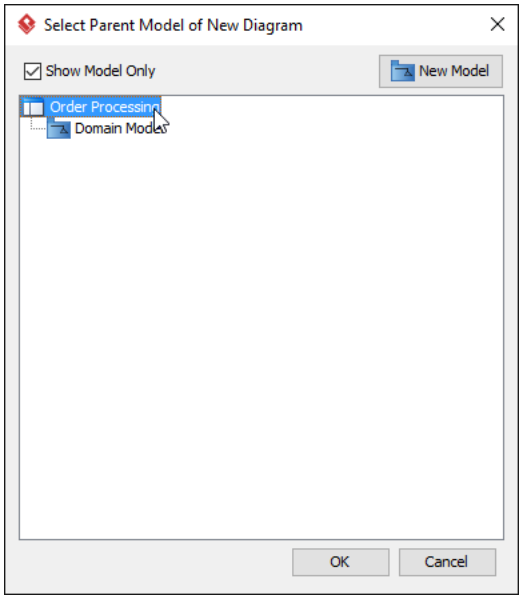
Creating an Implementation Class Model

Having completed the domain model, we are going to transcribe it into an implementation model.

1. Right-click on the diagram background and select **Utilities > Transit to New Diagram...** from the popup menu.

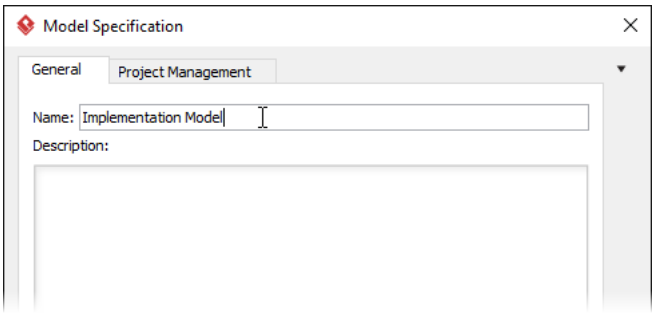


2. In the **Select Parent Model of New Diagram** window, select the project root node.



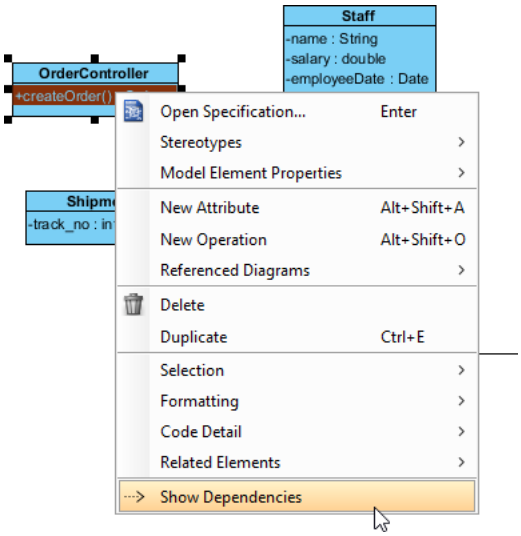
3. Click **New Model** at the top right of the window.

4. In the **Model Specification** window, enter *Implementation Model* as the model name. Click **OK** to confirm.

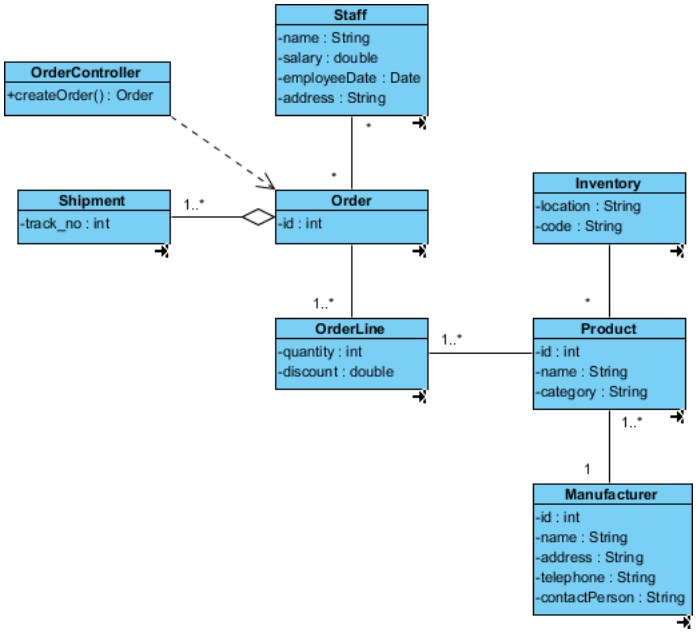


5. Keep *Implementation Model* selected, and click **OK** to confirm. This creates a new diagram named *Implementation Model*, which looks the same as *Domain Model*. What we are going to do is to refine the design to make it an implementation model.
6. Delete the *Customer* class by selecting it and pressing the delete key.
7. Rename class *Order line* to *OrderLine*.
8. Create a class *OrderController*.
9. Add an operation *createOrder() : Order* into *OrderController*.

10. Right-click on the operation *createOrder() : Order*, and select **Show Dependencies** from the popup menu.



Up to now, the implementation model should look like this:

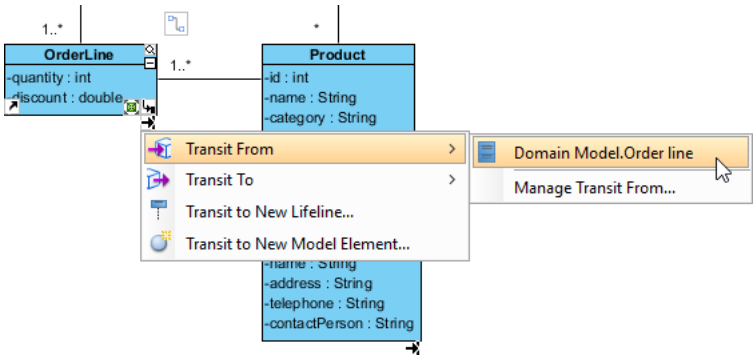


Using the Model Transitor

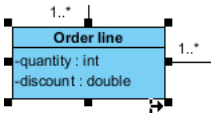
Let's refresh our memory on what we have changed for the class *OrderLine*.

1. Move your mouse pointer over the class *OrderLine*.

2. Click on the **Model Transitor** resource icon at the bottom right of the class shape and then select **Transit From > Domain Model.Order line** from the popup menu.



This populates the domain model, with the former version of class *OrderLine* selected, which was named *Order line*.

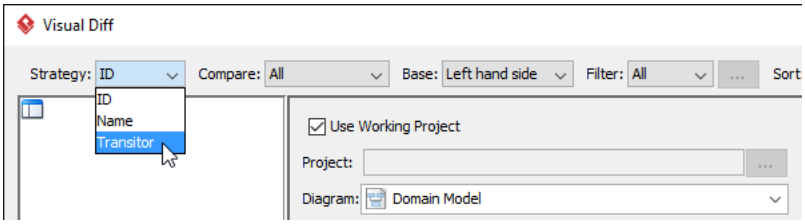


Comparing Models with Visual Diff

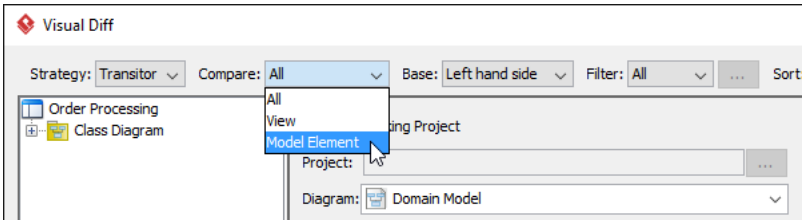
To identify all the refinements made in the implementation model, we need to use the Visual Diff tool.

1. Open Visual Diff by selecting **Modeling > Visual Diff** from the application toolbar.

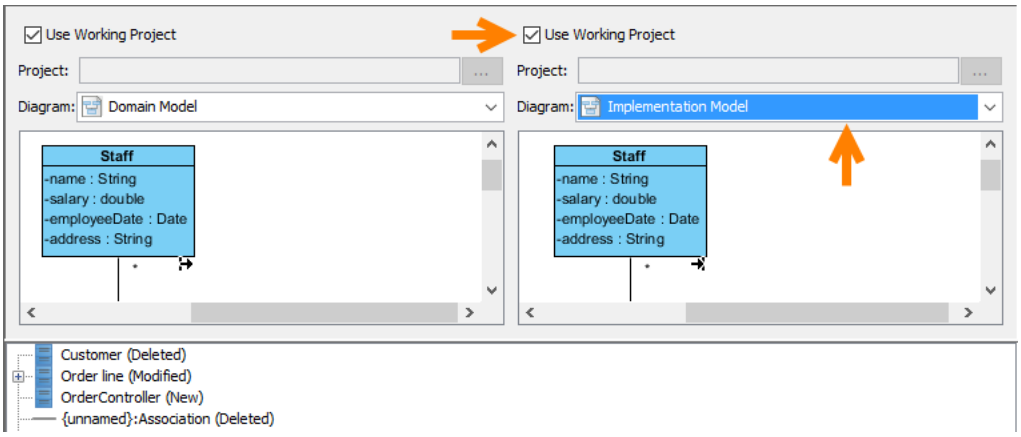
2. In the **Visual Diff** window, select **Transitor** as the **Strategy**.



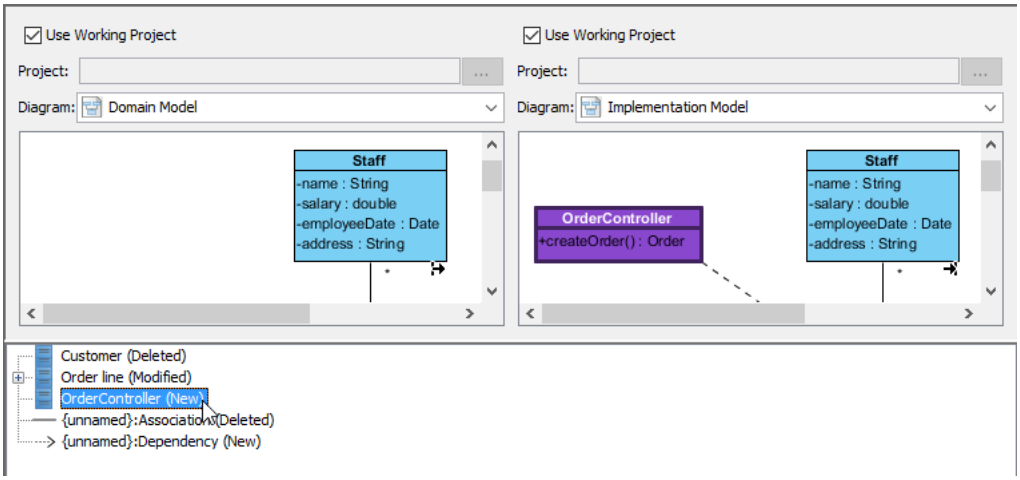
3. We want to know the changes in model element details rather than view details such as shape coordinates. Therefore, next to **Strategy**, we select **Model Element** to **Compare**.



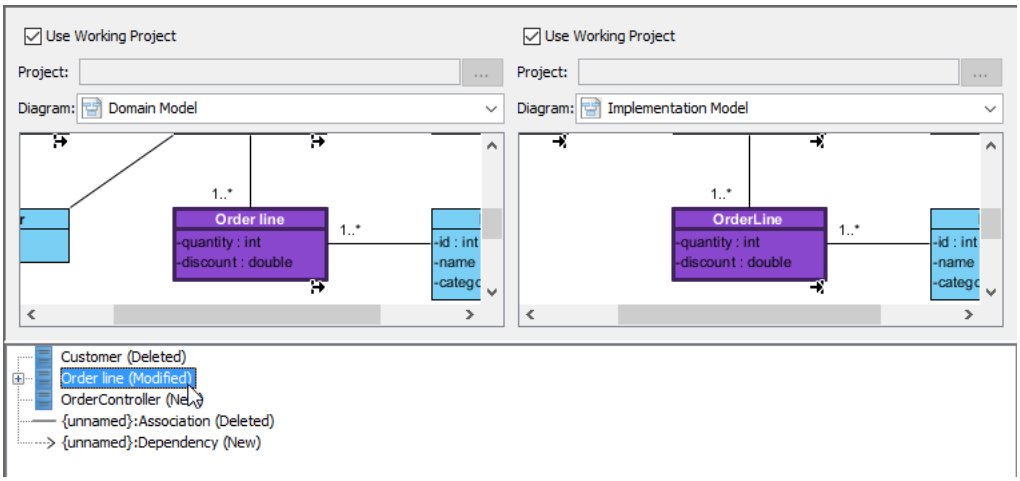
4. For the diagram pane on the right, keep **Use Working Project** selected and pick *Implementation Model* for the diagram to compare.



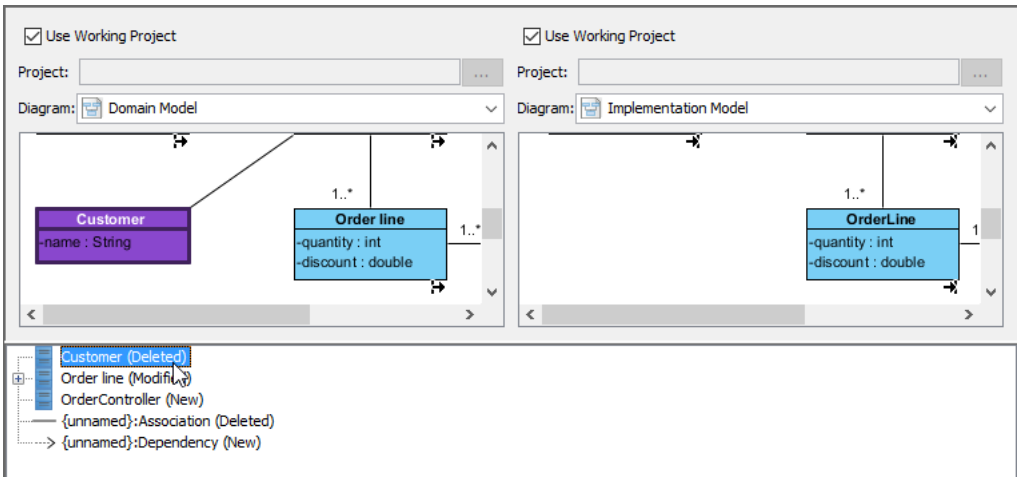
5. Now, we have obtained the differences between the domain (left-hand side) and implementation (right-hand side) models in the bottom pane. Click on the node **OrderController (New)**. From the tag (New) and from the diagram, we know that it is newly added in the implementation model.



6. Select and expand the node **Order line (Modified)**. We know that it has been renamed.



7. Select the node **Customer (Deleted)**. We can see that it no longer exists in the implementation model.



Visual Paradigm

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm tutorials
(https://www.visual-paradigm.com/tutorials/)