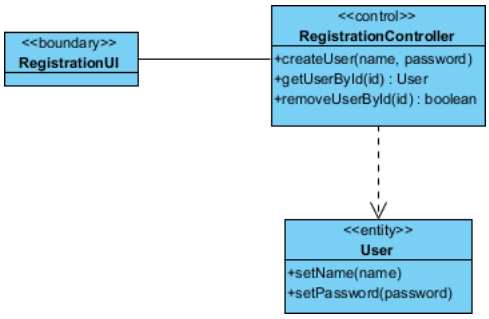
How to Draw UML Sequence Diagram?

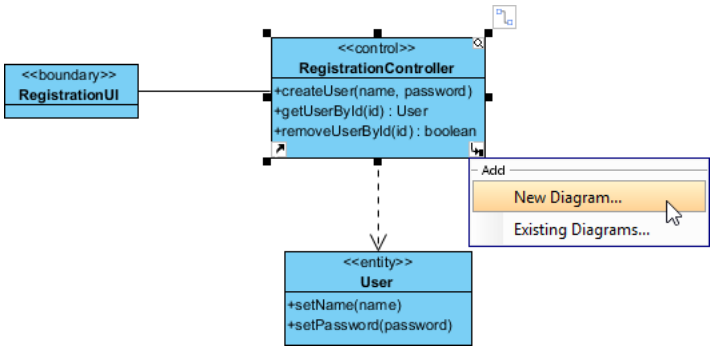
Written Date : March 16, 2016

Creating a Sequence Diagram from a Class

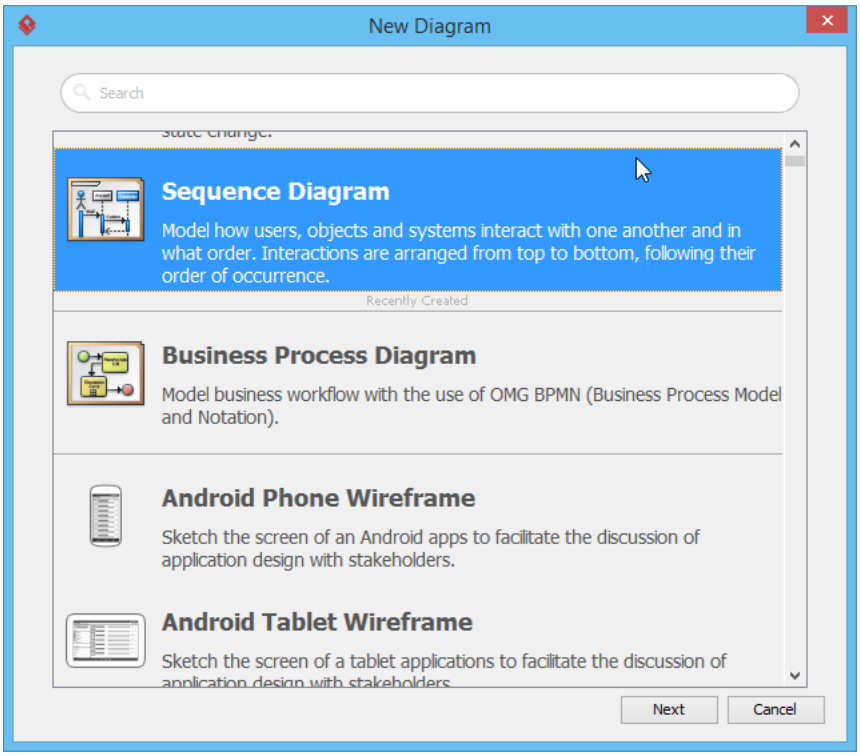
1. Download Simple-Registration.vpp. You can also find this file at the bottom of this tutorial.
2. Open the downloaded .vpp file in Visual Paradigm. To open a project, select **Project > Open** from the application toolbar.
3. Open the class diagram *Registration*. Study the diagram content. We have three classes: *RegistrationUI*, *RegistrationController*, and *User*.



4. Now, we want to model the interaction between object instances of these classes at runtime. Since the controller class is responsible for controlling the registration process, add a sub-sequence diagram from it. Move the mouse pointer to *RegistrationController*. Click on the resource icon at the bottom right corner and select **New Diagram...** from the popup menu.

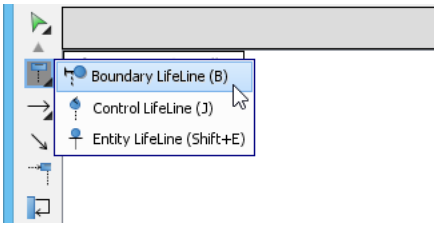


5. In the **New Diagram** window, select **Sequence Diagram** and click **Next**. Keep the diagram name as provided and click **OK** to confirm. This creates an empty UML sequence diagram.

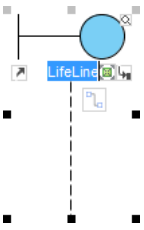


Drawing the Sequence Diagram

1. Select **Boundary Lifeline (B)** from the diagram toolbar.

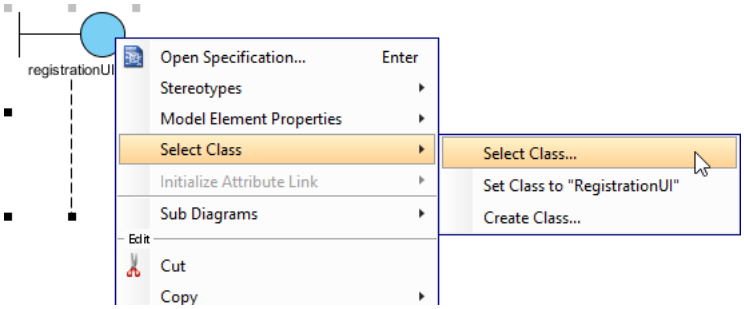


2. Click on the diagram to create a boundary lifeline.

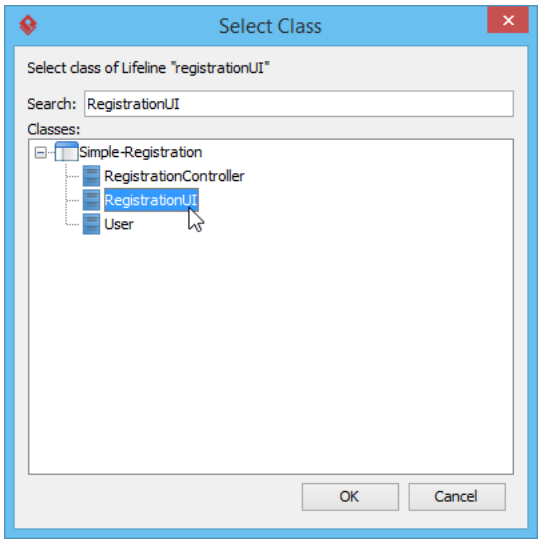


3. Enter *registrationUI* as the name of the lifeline and then press **Enter** to confirm.

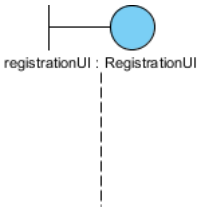
4. Right-click on the lifeline and select **Select Class > Select Class...** from the popup menu.



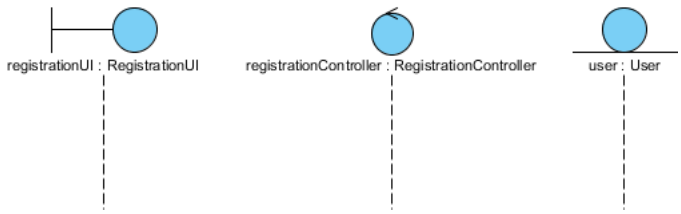
5. In the **Select Class** window, select the *RegistrationUI* class and then click **OK** to confirm.



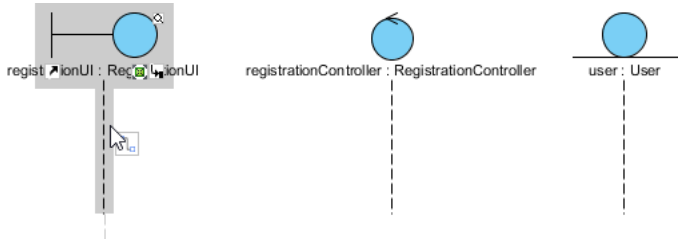
After that, the lifeline will look like the following.



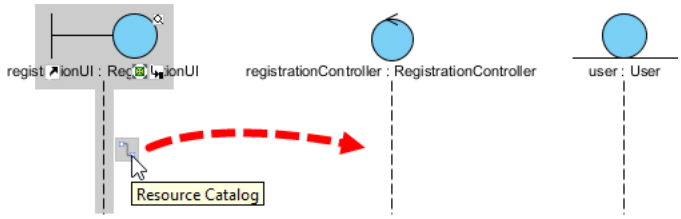
6. Create a **Control Lifeline** *registrationController : RegistrationController* and an **Entity Lifeline** *user : User*. Don't forget to select the appropriate classes for them. The diagram will look like the following.



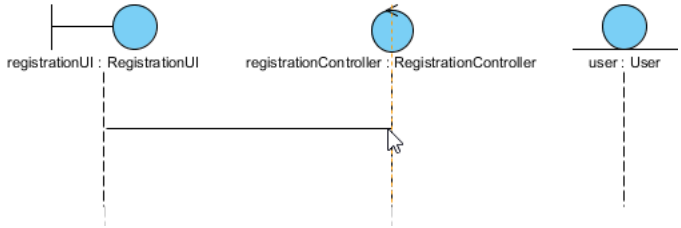
7. Let's model the method invocations between lifelines. Move the mouse pointer over lifeline *registrationUI*.



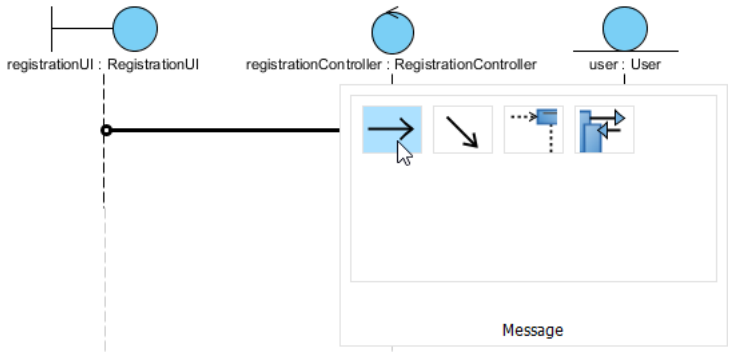
8. Click on the **Resource Catalog** resource and drag it out.



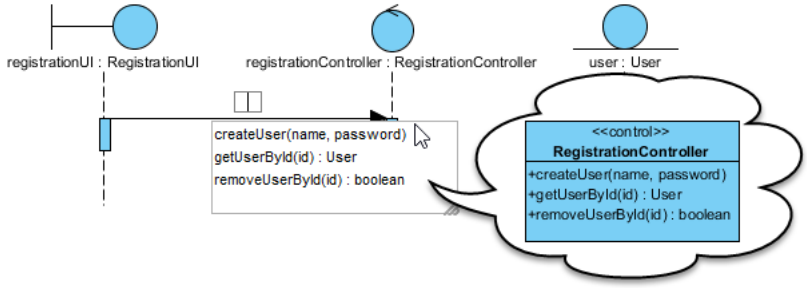
9. Move to lifeline *registrationController* and release the mouse button.



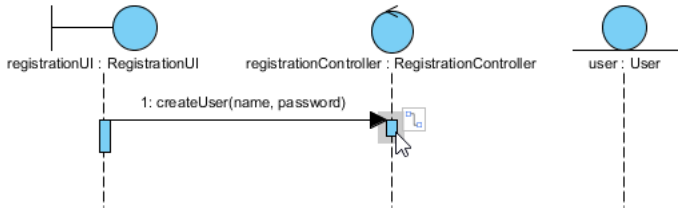
10. Select **Message** from the Resource Catalog.



11. This pops up a list of names that you can choose for the new sequence message. You can see that those are operations of *classRegistrationController*. Select *createUser(name, password)*.



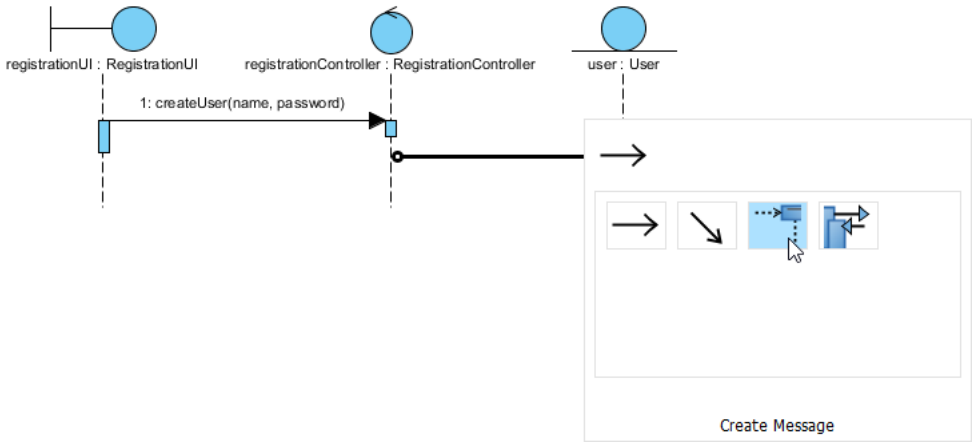
12. Relate lifeline *registrationController* and user. We say that *registrationController* creates the user lifeline. Therefore, we need to relate them with a create message. Move the mouse pointer over the activation in lifeline *registrationController*.



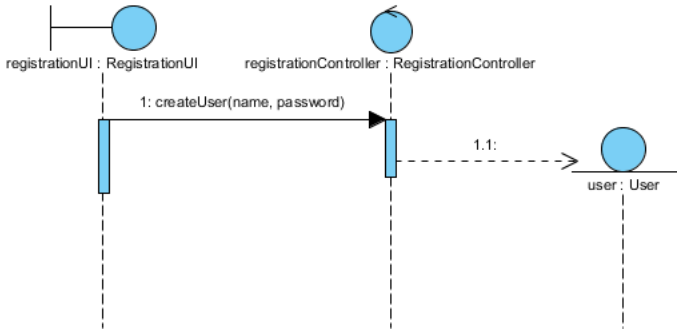
13. Click on the resource **Resource Catalog** and drag it out.

14. Release the mouse button on the lifeline *user*.

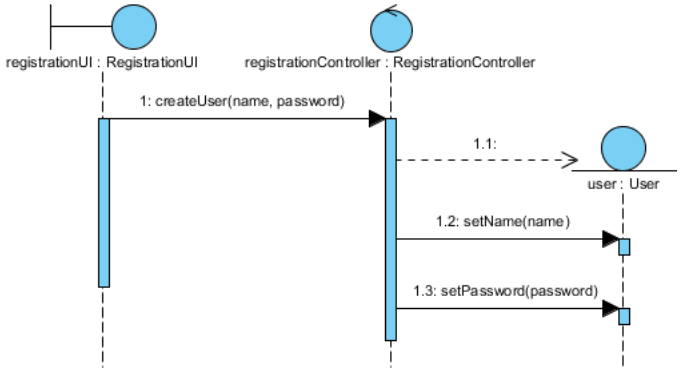
15. Select **Create Message** from the Resource Catalog.



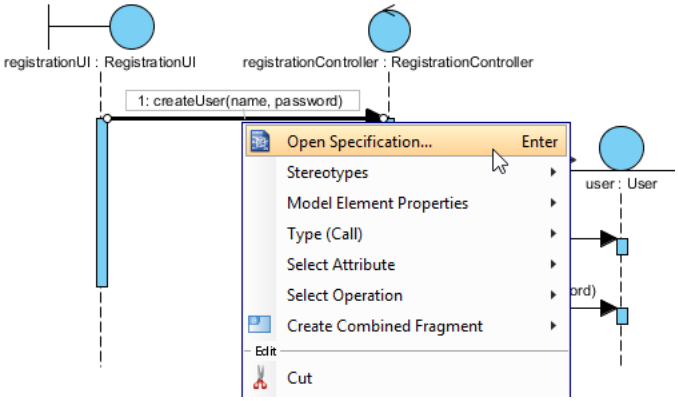
A create message is created. Your diagram should look like this:



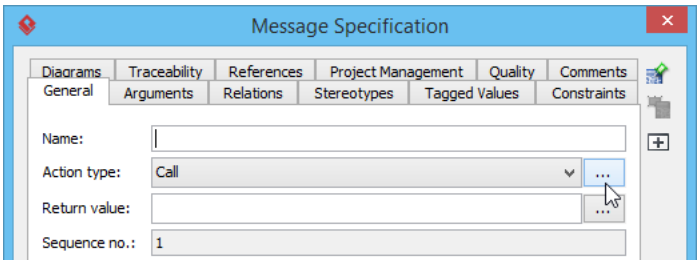
16. Create messages *setName* and *setPassword* from lifeline *registrationController* to user. Up to now, the diagram becomes:



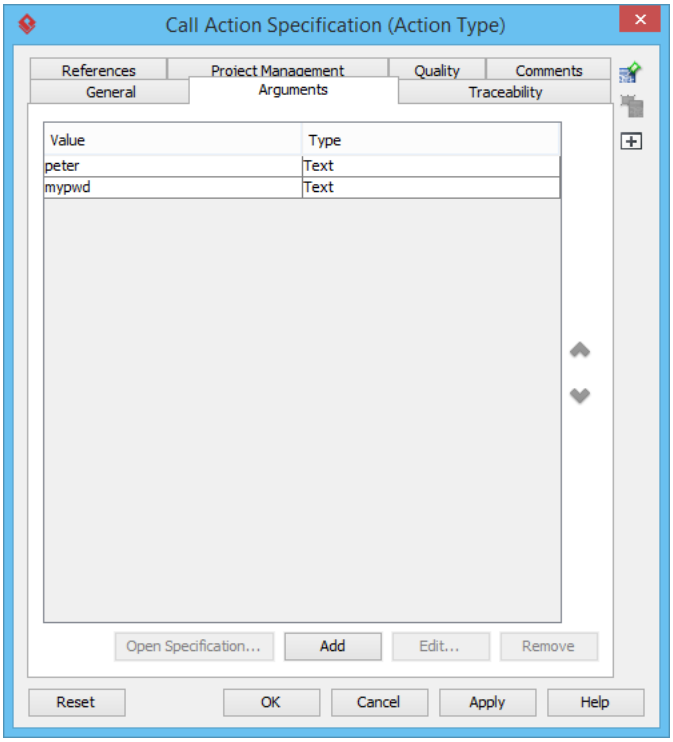
17. You can also specify the arguments of operations. Take the message *createUser(name, password)* as an example. Right-click on it and select **Open Specification...** from the popup menu.



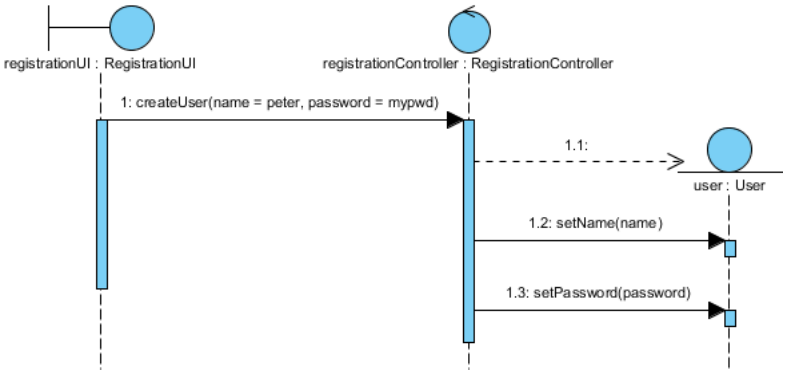
18. Edit the action type property by clicking on the button with the dotted caption, next to **Action type**.



19. In the **Call Action Specification** window, click **Add > Text...** to add an argument. In this example, click **Add > Text...** again to add argument *peter*. Click **Add > Text...** again to add argument *mypwd*. Note that the two arguments are actually referring to the two parameters given by the operation. If you add the third argument here, it will be ignored (as there are only two operations defined).



20. Click **OK** to close the windows and go back to the diagram. The arguments are added and presented on the diagram. Finally, the diagram becomes:



Resources

1. What is Sequence Diagram?
2. Simple-Registration.vpp

Related Links

- More UML sequence diagram toolset Full set of UML tools and UML diagrams

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

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