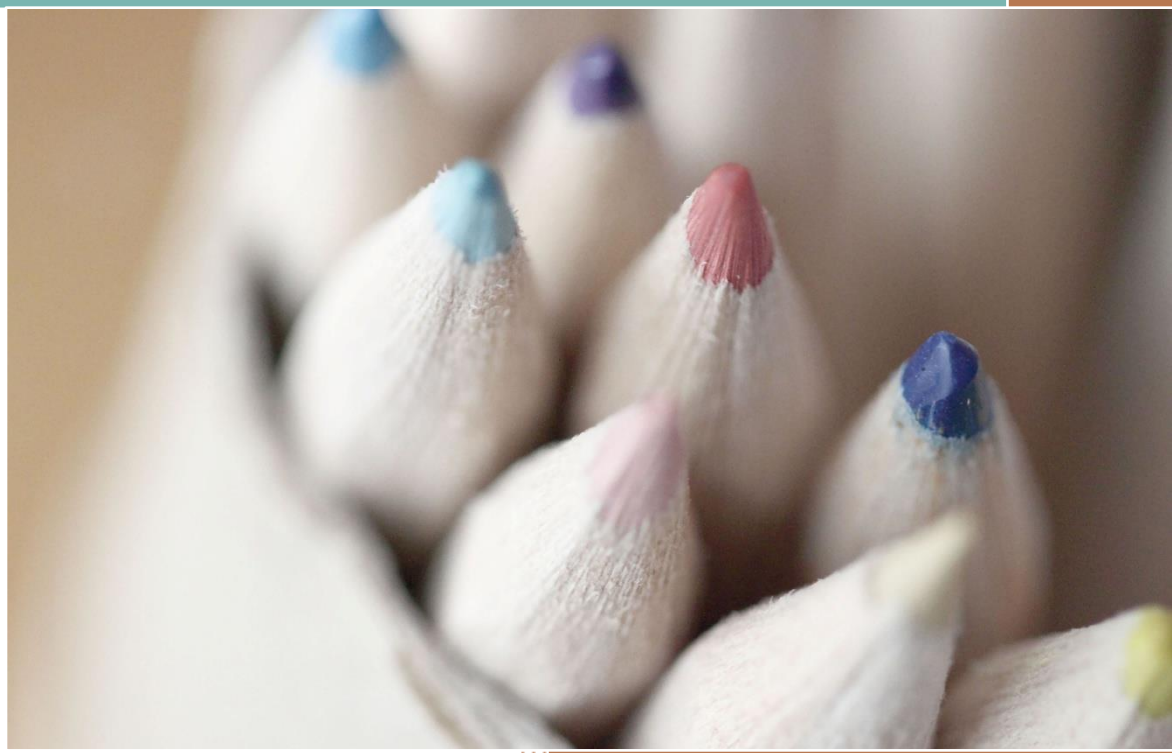
User Manual
I-MHERE 2012

Integrated Software Development Tool in Behavior Tree



Agung Pratama, Ardi, Chairunnisa Atimas, Emerson C. Simbolon, Ikhsanul Habibie

Fakultas Ilmu Komputer
Universitas Indonesia
2012

Table of contents

Features of GraphBT and how to use it

Add Component

Specifying a new Behavior Tree component in GraphBT can be performed by using add component feature. User can specify a new BT component via Add Component button provided in the wizard that is used to create a new BT node. After the button is pressed, new wizard will appear and user can specify the name and reference of the created component. Another way to add new component is by clicking Add New Component button located in the toolbar. In this way, we can define a new component without necessarily creating a new node.

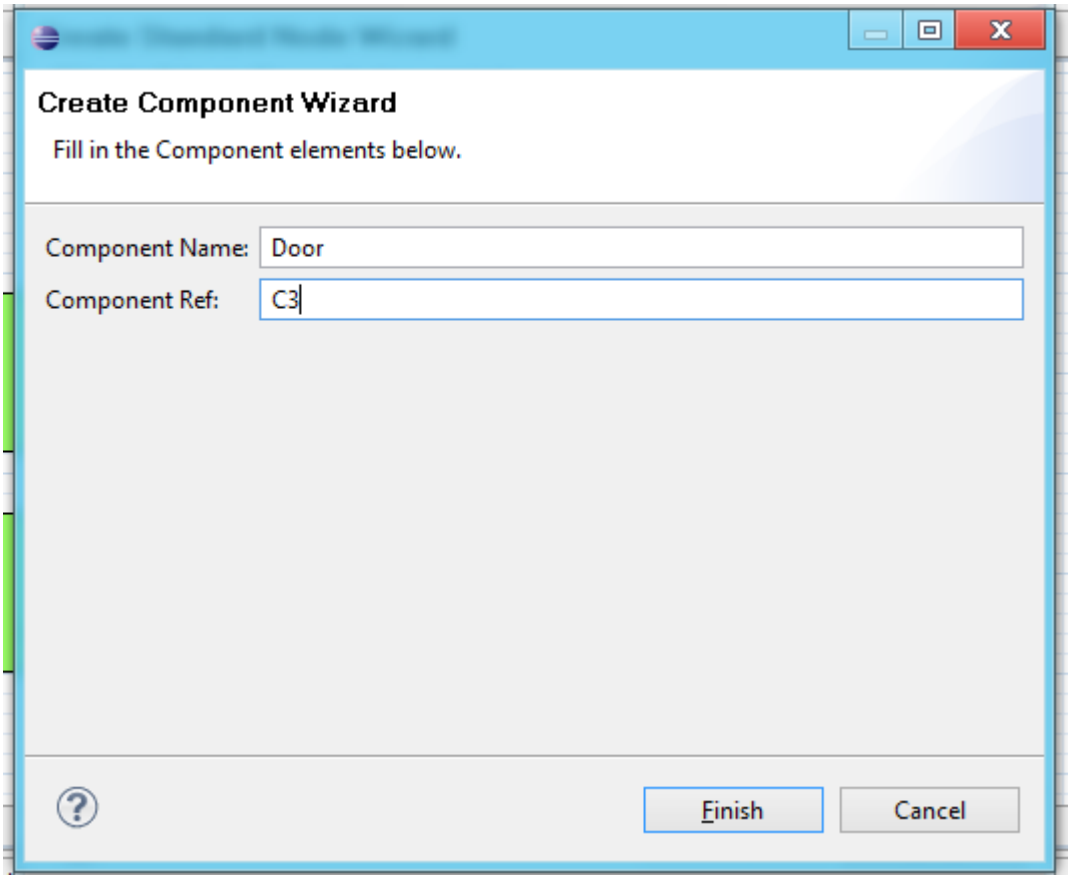


Figure 1 Create Component wizard

Add Behavior

Adding new behavior can be accomplished after the respective component has been specified. This can be performed in create new BT node wizard. If the node is already associated with a component, then user can add new behavior by pressing Add New Behavior button. A new wizard will be invoked so that the user can specify the reference and name of the behavior. Another way to this is by using Manage Component button in the toolbar which its use will be explained later.

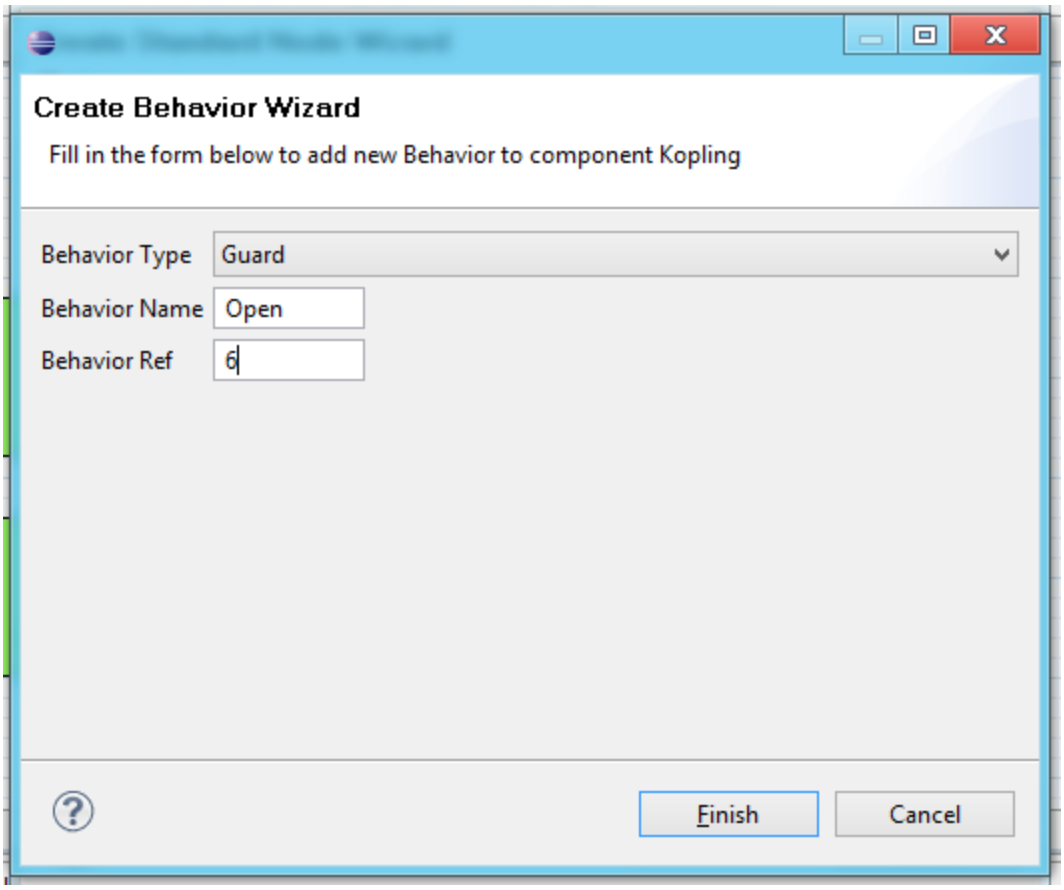


Figure 2 Add Behavior wizard

Manage Components

BT Components and their respective behaviors can be managed using Manage Component Wizard which can be invoked by pressing Manage Component button in the toolbar. In this wizard, user can add new component, delete existing component, and modify a component name or reference. When a component is selected, the respective behavior list will be shown and can also be modified with the same fashion as their component.

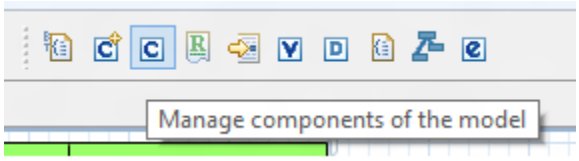


Figure 3 Manage Components icon in the toolbar

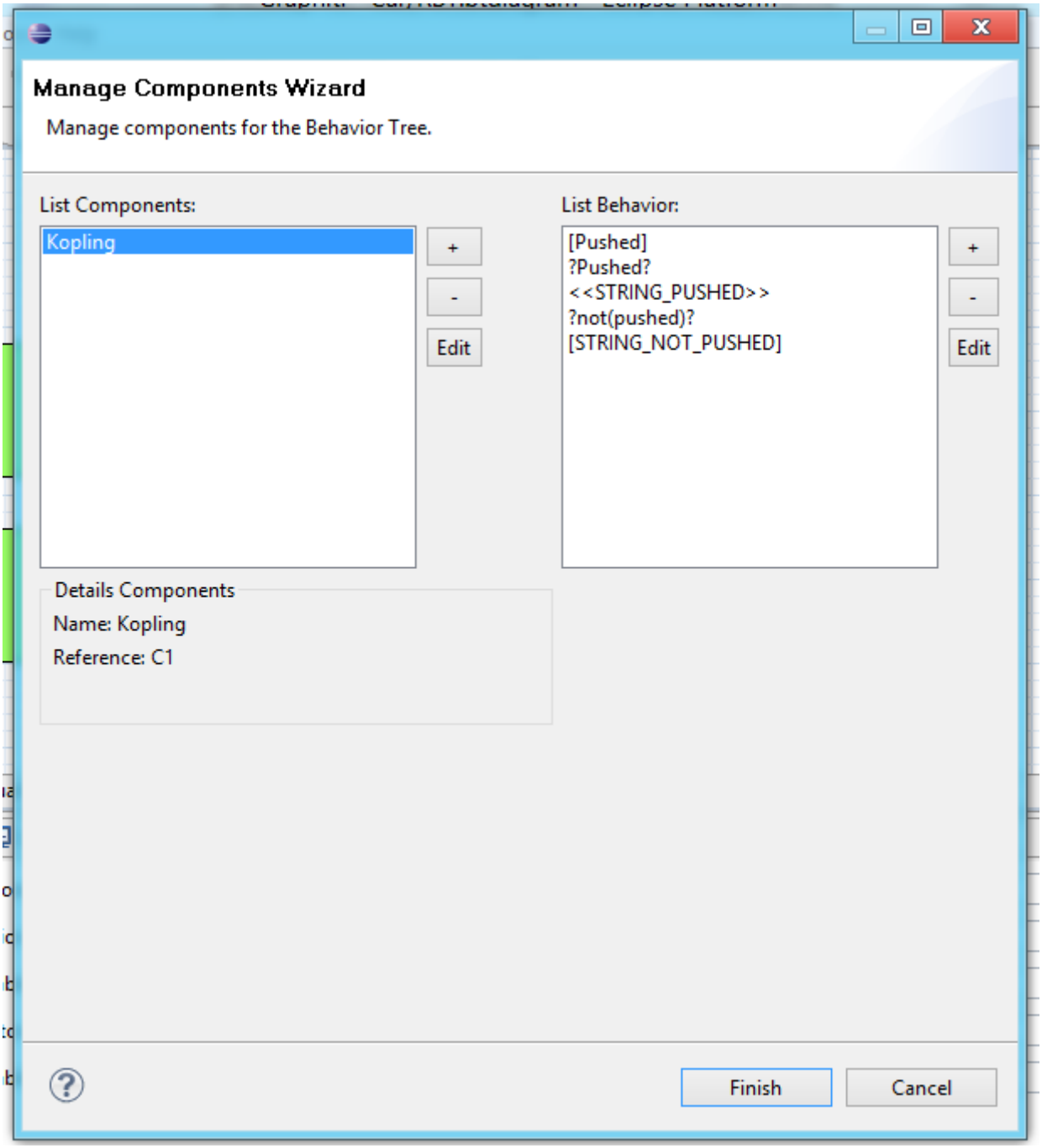


Figure 4 Manage Components wizard

Add Requirement

Adding new requirement can be performed in the same fashion with add component feature. It can be done by pressing Add Requirement button of Create BT Node Wizard. A new wizard for managing requirement will be invoked and user can add, modify, and delete requirements. Choose button Add New Requirement in this wizard and then Requirement name, Requirement reference, and description text field will appear.

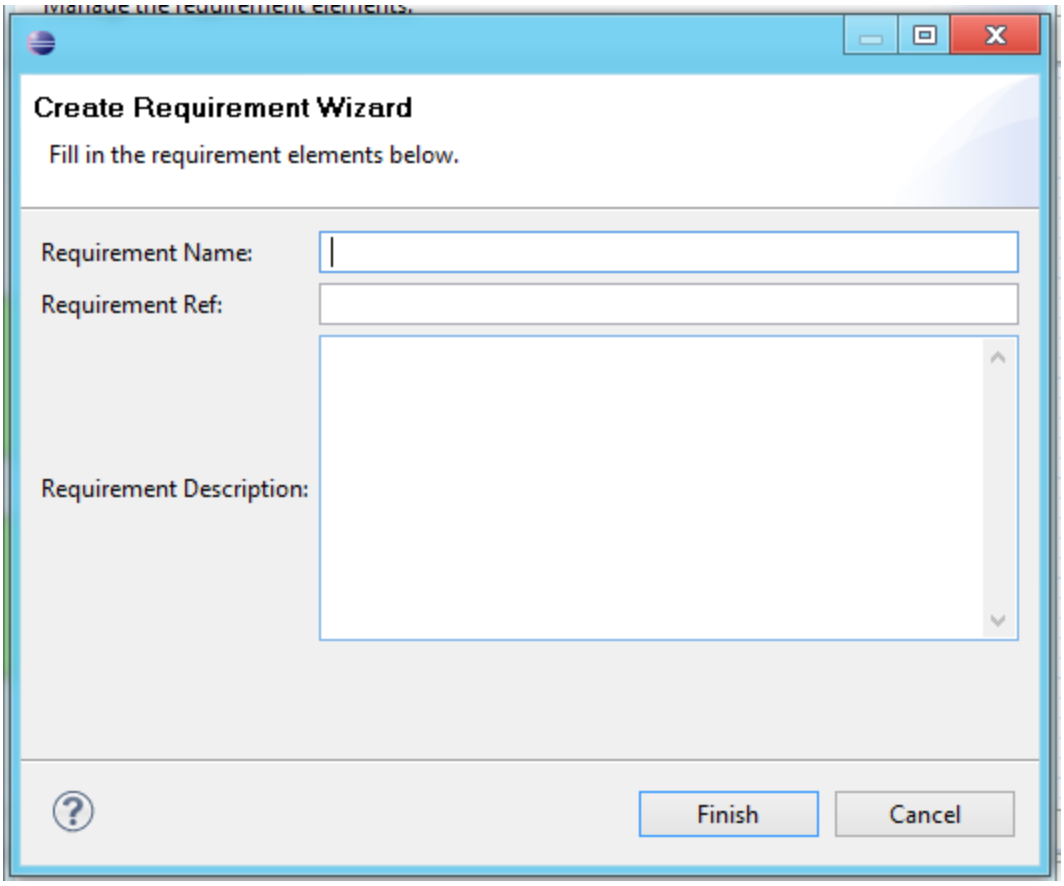


Figure 5 Create Requirement wizard

Manage Requirement

User can add, modify, and delete requirements using Manage Requirement feature. This can be performed by clicking Manage Requirement button in the toolbar. Afterward, Manage Requirement wizard will appear.

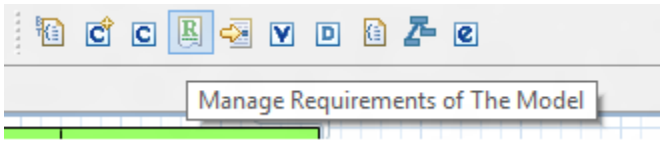


Figure 6 Manage Requirements icon in the toolbar

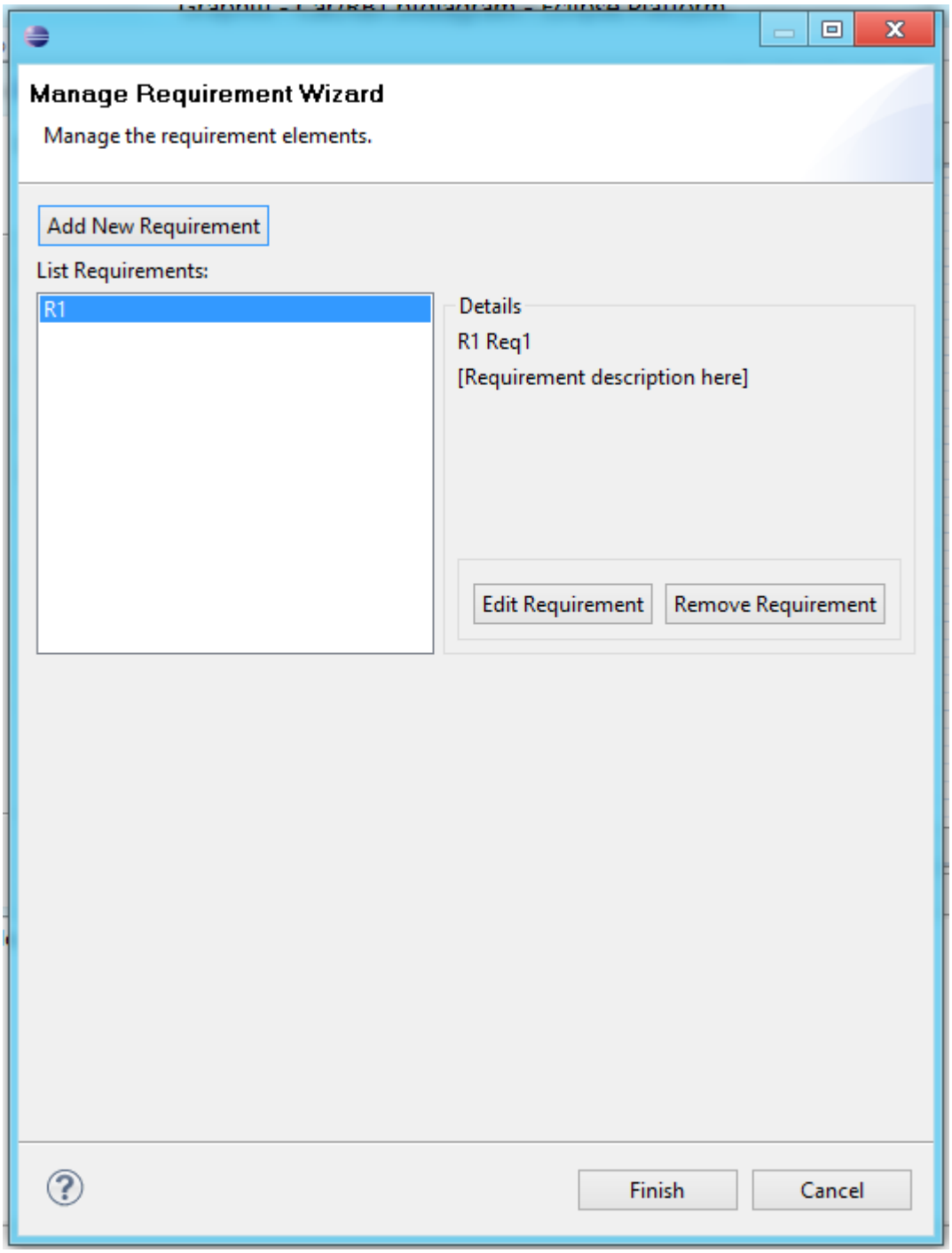


Figure 7 Manage Requirement wizard

Edit Requirement

User can edit any created Requirements by selecting a Requirement from selection list and press Edit Requirement button afterward.

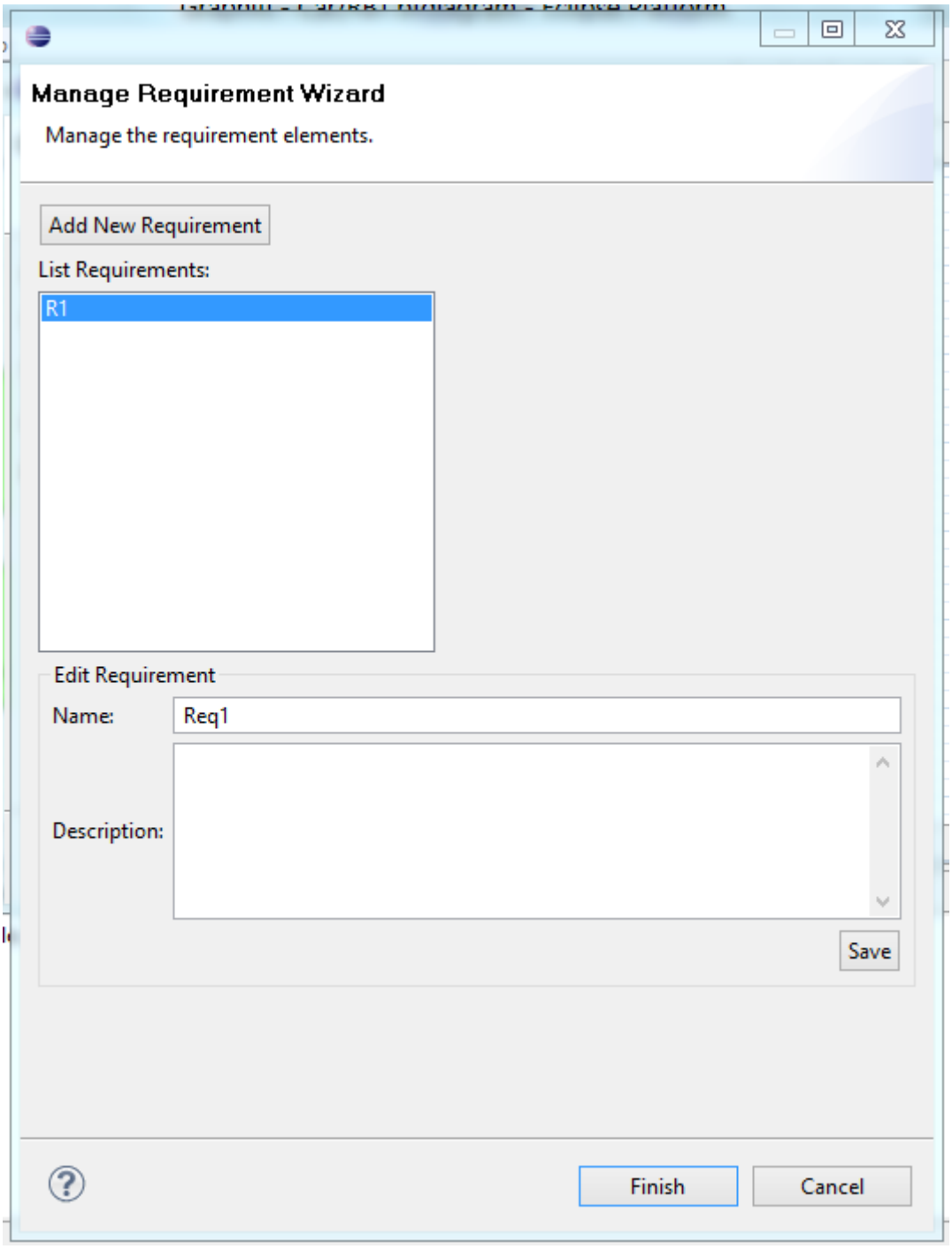


Figure 8 Edit Requirement view in Manage Requirement Wizard

Add Behavior Tree Node feature

We can add new BT node to the clipboard by clicking General BT Node label in the palette. Add New BT Node wizard will appear. User can select the Component, Behavior, Traceability Link, Traceability Status, and Operator for the node using five available combo boxes provided that the Requirement, Component, and Behavior have been added via their respective add features. In

order to complete this Add BT Node feature, at least a Component and one respective Behavior of the Component have to be selected from the combo box.

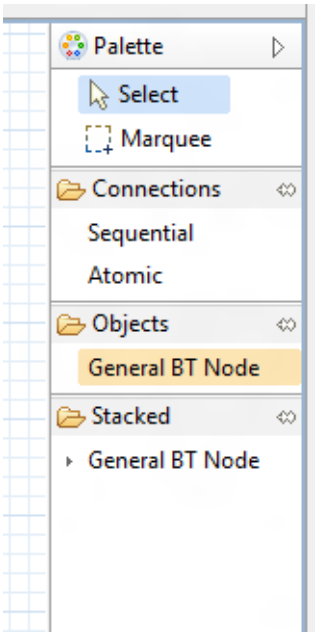


Figure 9 Select General BT Node label in the palette to create a new BT Node

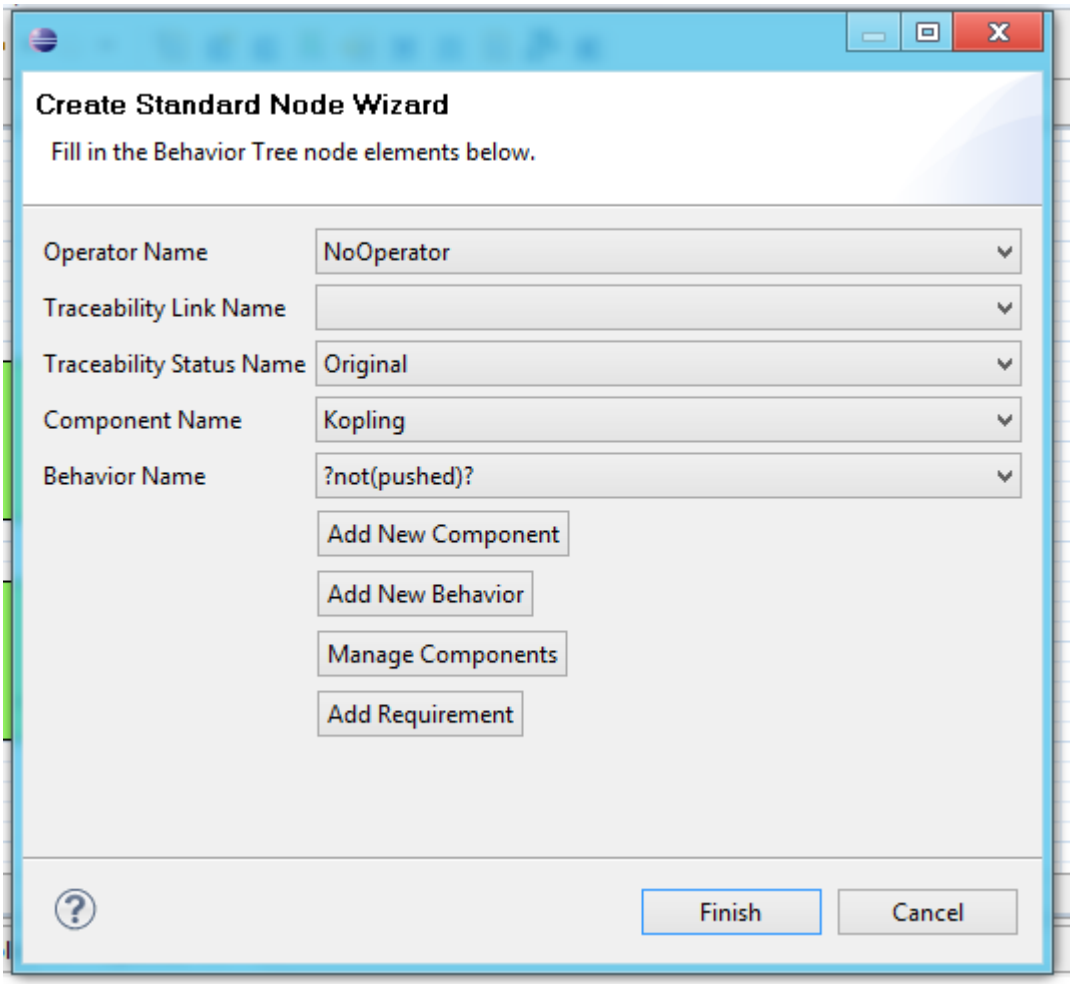


Figure 10 User can specify a new BT Node in Create Standard Node wizard

Add Behavior Tree Connection feature

There are two types of connection in Behavior Tree specification: sequential and atomic connection. Both of these connections can be invoked from the connection list in the palette. To connect two BT nodes either by sequential or atomic connection, first select the connection from the palette, and then click a BT node that is going to be connected, and finally click another BT node.

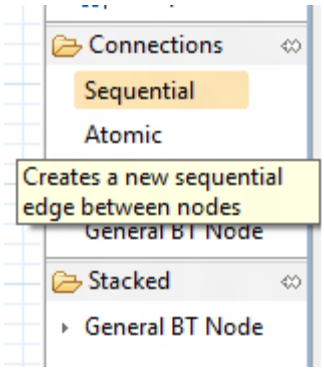


Figure 11 to create an edge between nodes, click Sequential or Atomic label from the palette, then click both nodes

To create branching connection, simply connect an already connected BT Node to another BT node. After the new BT node target is connected, a selection wizard will appear to select the branch type; either it is parallel or alternative branch.

Validate Behavior Tree

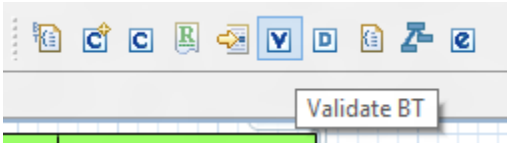


Figure 12 Validate BT icon in the toolbar

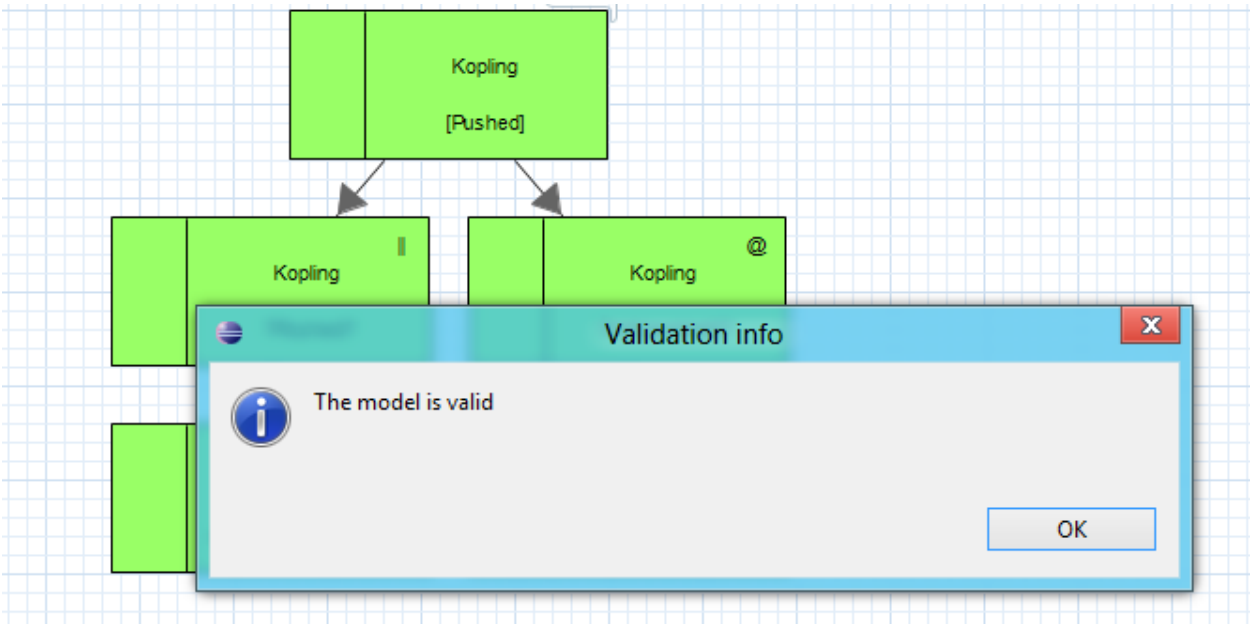


Figure 13 after validating the BT, a message will appear to inform whether the BT is valid or not

After creating a Behavior Tree, user can check whether the design is valid or not by pressing Validate BT button provided in the toolbar. If the design is not valid, an error message will appear and inform the user about the error. Otherwise, the message will inform that the design is valid.

Generate BT Code

After the Tree is added to the diagram editor, user can generate the .bt code by pressing Generate BT Code button in the toolbar. The generated BT code will then appear in the Project Explorer.

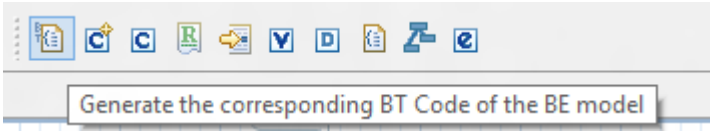


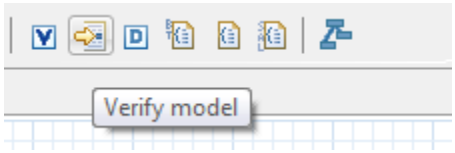
Figure 14 Generate BT code icon in the toolbar

Generate Java Code

Verify Behavior Tree

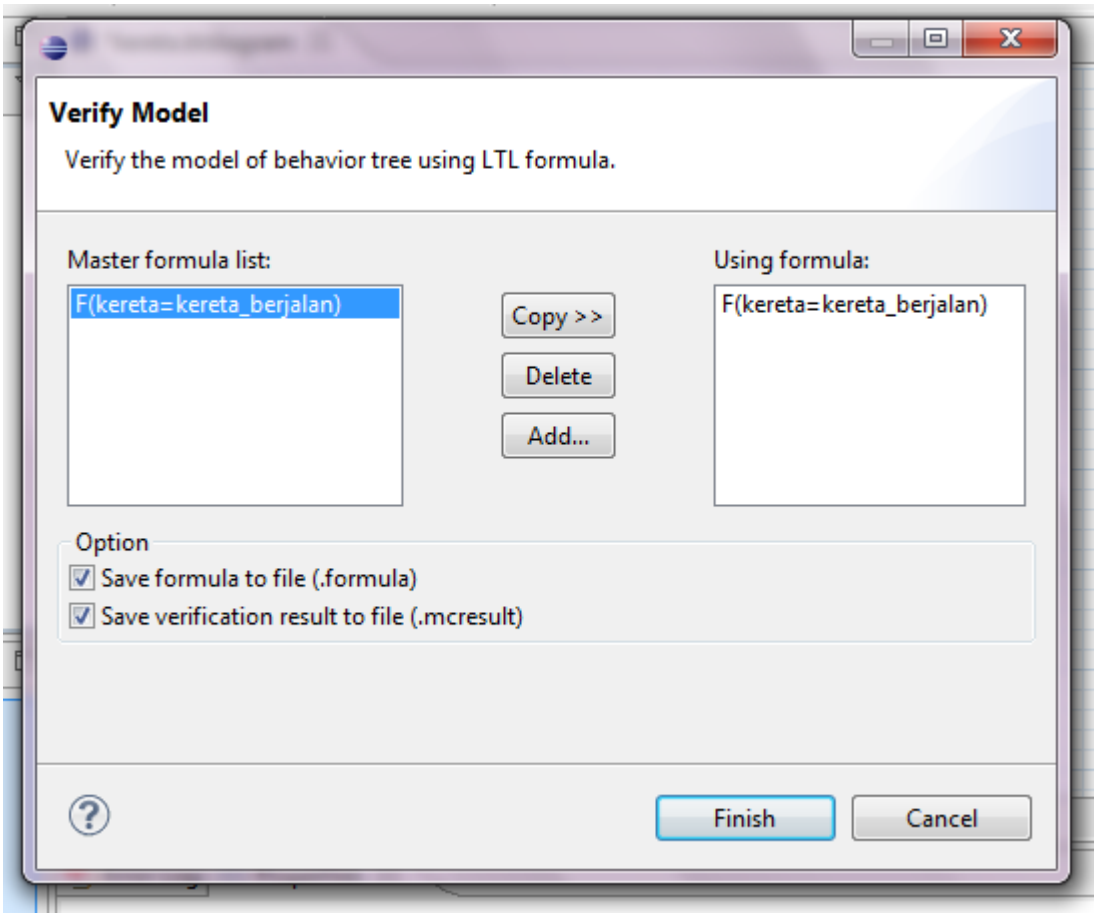
This feature is used to verify modeled requirement in behavior tree. For this need, LTL formula needed to check whether the property of system fulfilled. If fulfilled, "PROVEN" output will be produced. Otherwise, counter-example which shows the path of the system that causes the property is not fulfilled will be produced.

To verify behavior tree, first, user clicks “Verify model” button.



Verify model icon in the toolbar

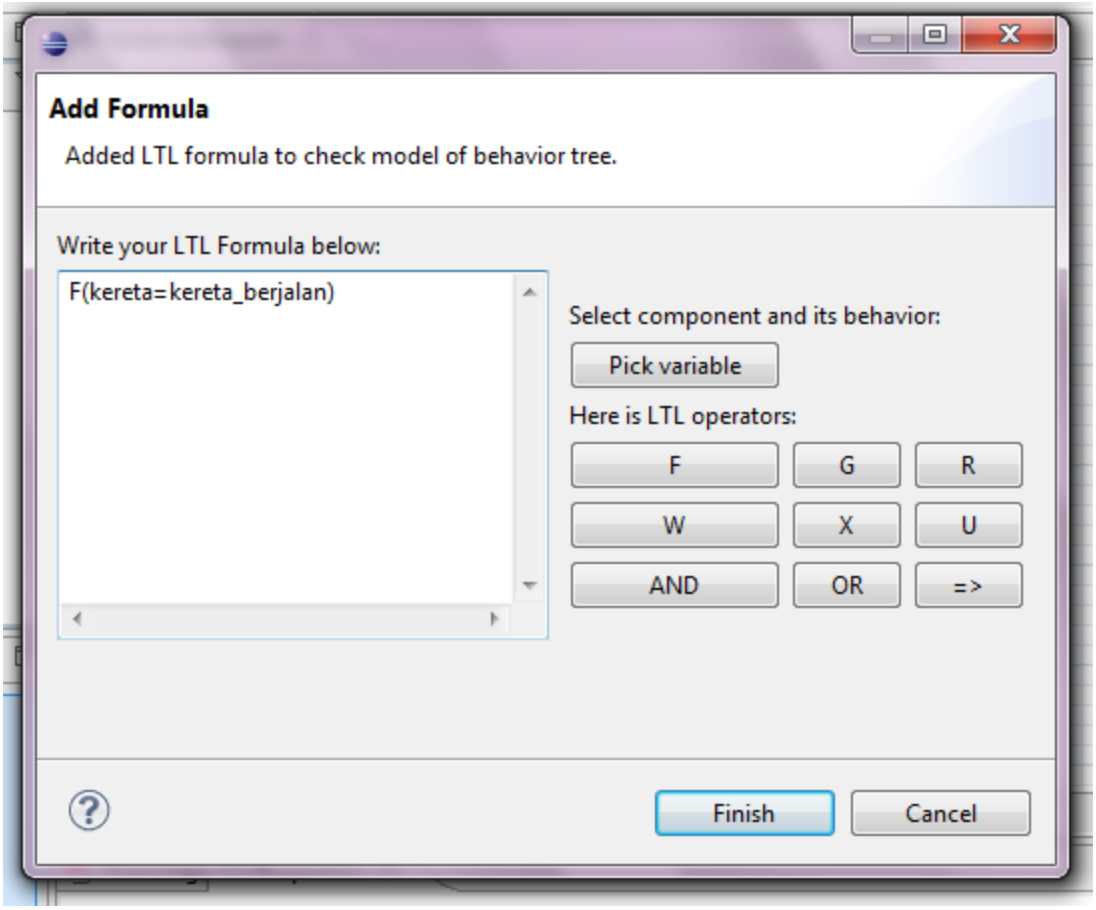
Then “Verify model” window will be appears. Then user can add LTL formula based on behavior tree from this window by clicking “Add…” button. The LTL formula acts as rule that must be fulfilled by behavior tree. This formula is created based on behavior tree components and its behavior. User can save the formulas or the verification result to file in the Project Explorer by tick check list on “Verify model” window.



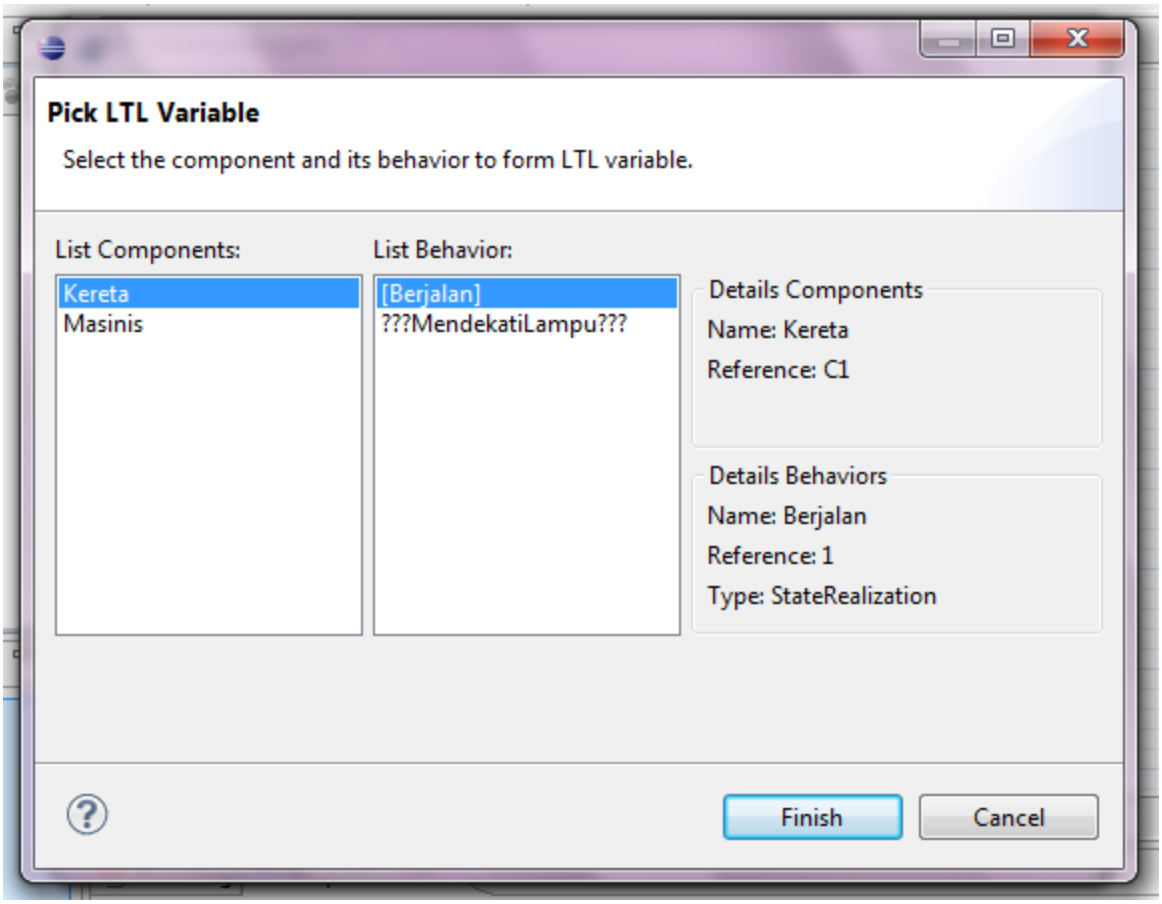
View of Verify Model wizard

When finish button performed, verifying model of behavior tree is conducted. It will produce file output with extension *.mcresult. When the property is not fulfilled, counter example represented bttrace format is produced as output.

User can add formula by clicking “Add...” button in the “Verify model” window. It will invoke “Add formula” window. Here user writes the desired LTL formula. “Add formula” window contains LTL formula operator buttons and “Pick variables” button to select appropriate LTL variable based on behavior tree specification.



View of Add Formula window



View of Pick LTL Variable window

Debugging Tools

User can also debug an already created BT diagram by clicking Debug BT Diagram button in the toolbar. A new perspective will be opened. In this perspective, user can animate the Behavior Tree execution and monitor the execution sequence. This feature will also inform user about the error found in the Behavior Tree design.

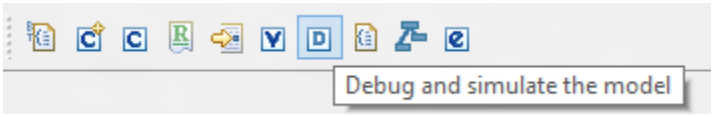


Figure 15 Debug and simulate BT icon in the toolbar

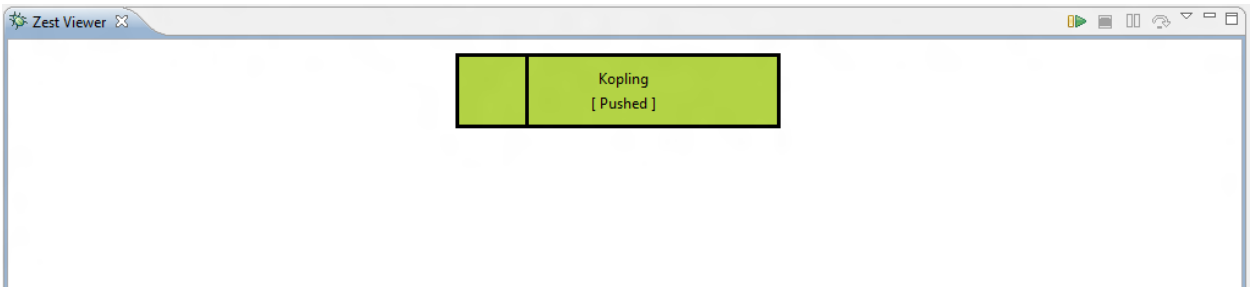


Figure 16 View of debugger and simulation tool

Figure 17 state viewer for Debugger

Case Study

In this section we will show how to create a behavior tree from a real requirement using GraphBT. Suppose we want to create an application called Mouse Mover. Mouse mover is an application to move the mouse pointer. Sometimes, a computer setting will caused computer screen to be turned off if there is no interaction. This program will cause the mouse to patrol to a certain location, so then the computer always has interaction, hence the monitor won't be turned off.

The requirement of Mouse Mover is shown below:

R1 The screen shows instruction to choose between three options. Option one, move the mouse to one location only. Option two, patrol the mouse. And option three, to exit the program.

R2 If the user chooses option 1, then the user is asked to input a coordinates in (X,Y). After that, the mouse is moved to (X,Y) and the screen displays the instruction again.

R3 If the user chooses option 2, then the user is asked to input two coordinates in (X1,Y1) and (X2,Y2). After that, the mouse moved from (X1,Y1) to (X2,Y2) repeatedly. The mouse will stop moving when the user types any character to the screen.

In order to create the behavior tree from Mouse Mover, we will first import all the requirements.

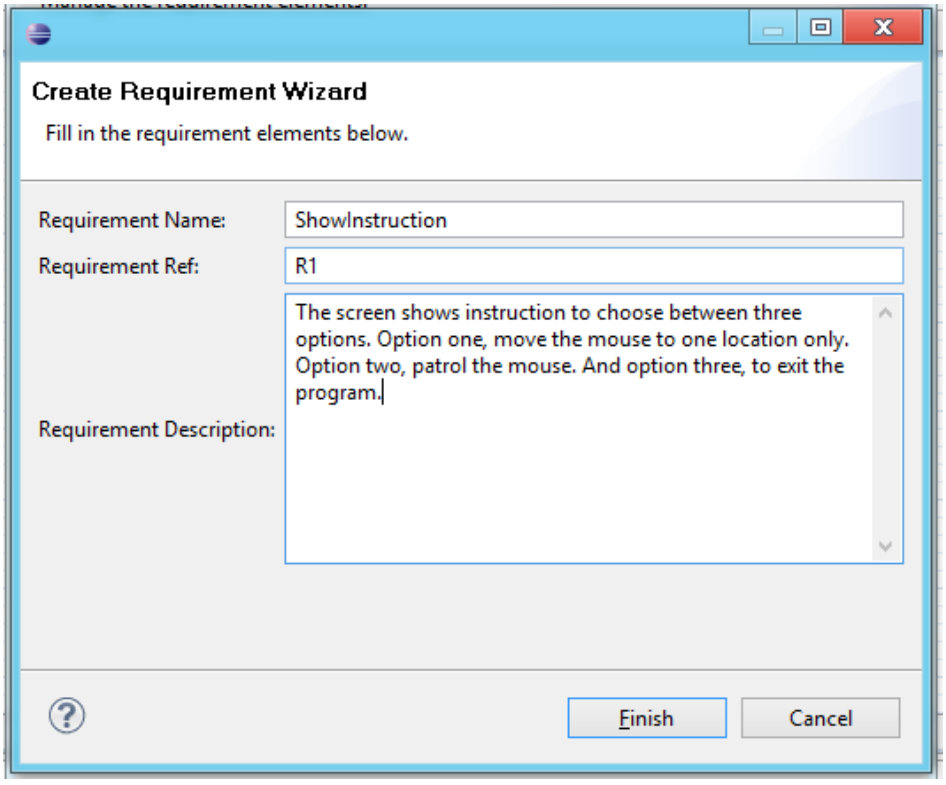


Figure 18 Create requirement ShowInstruction for Mouse Mover

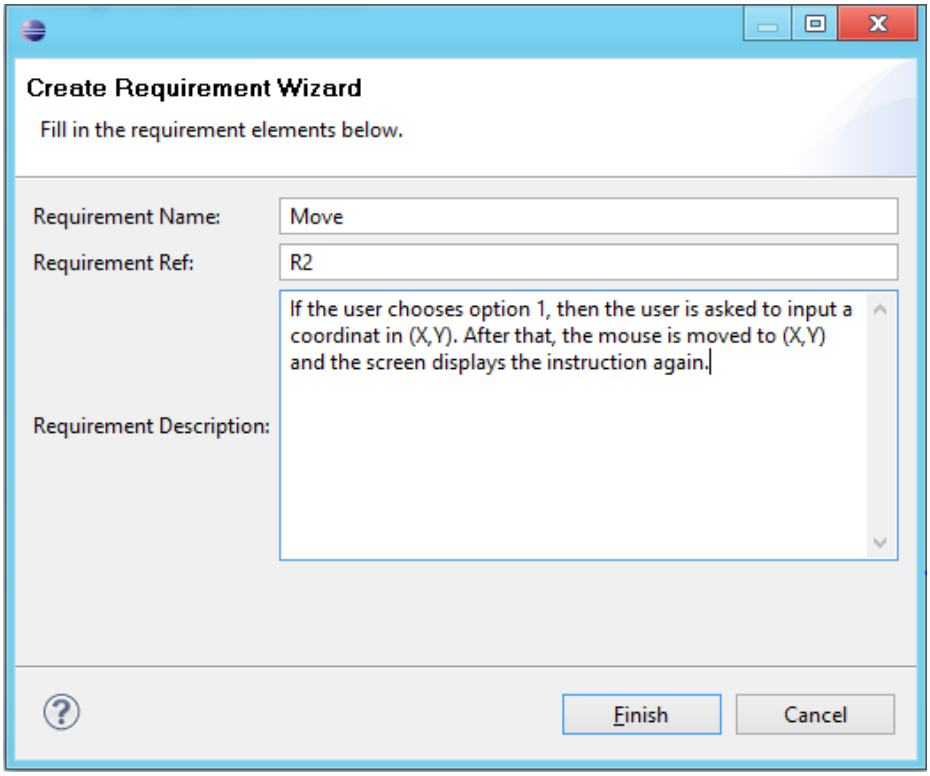


Figure 19 Create requirement Move for Mouse Mover

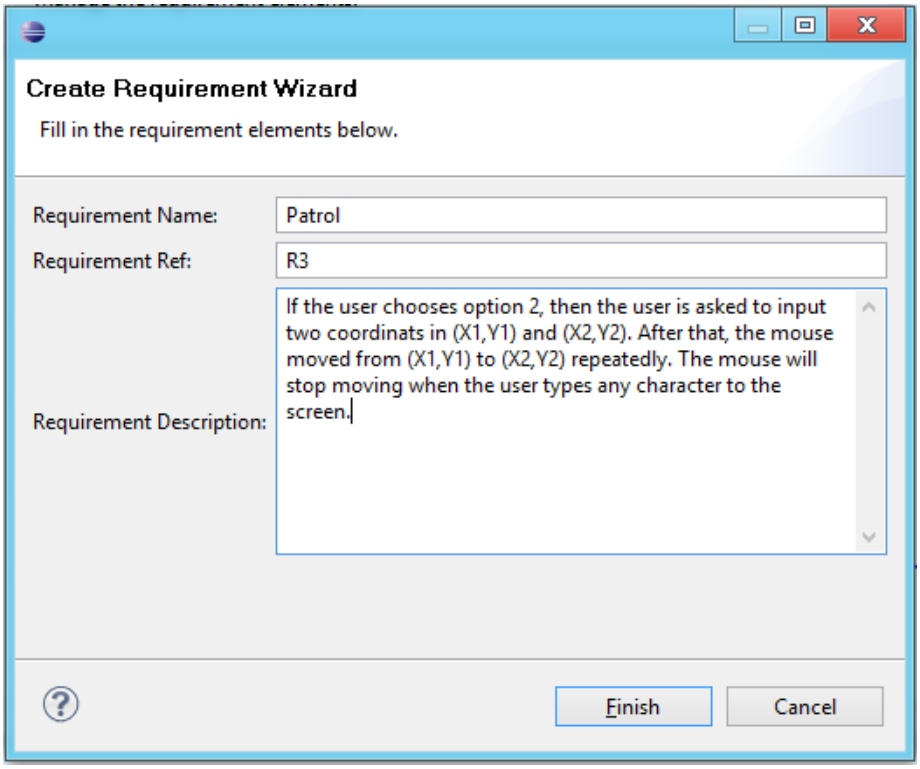


Figure 20 Create requirement Patrol for Mouse Mover

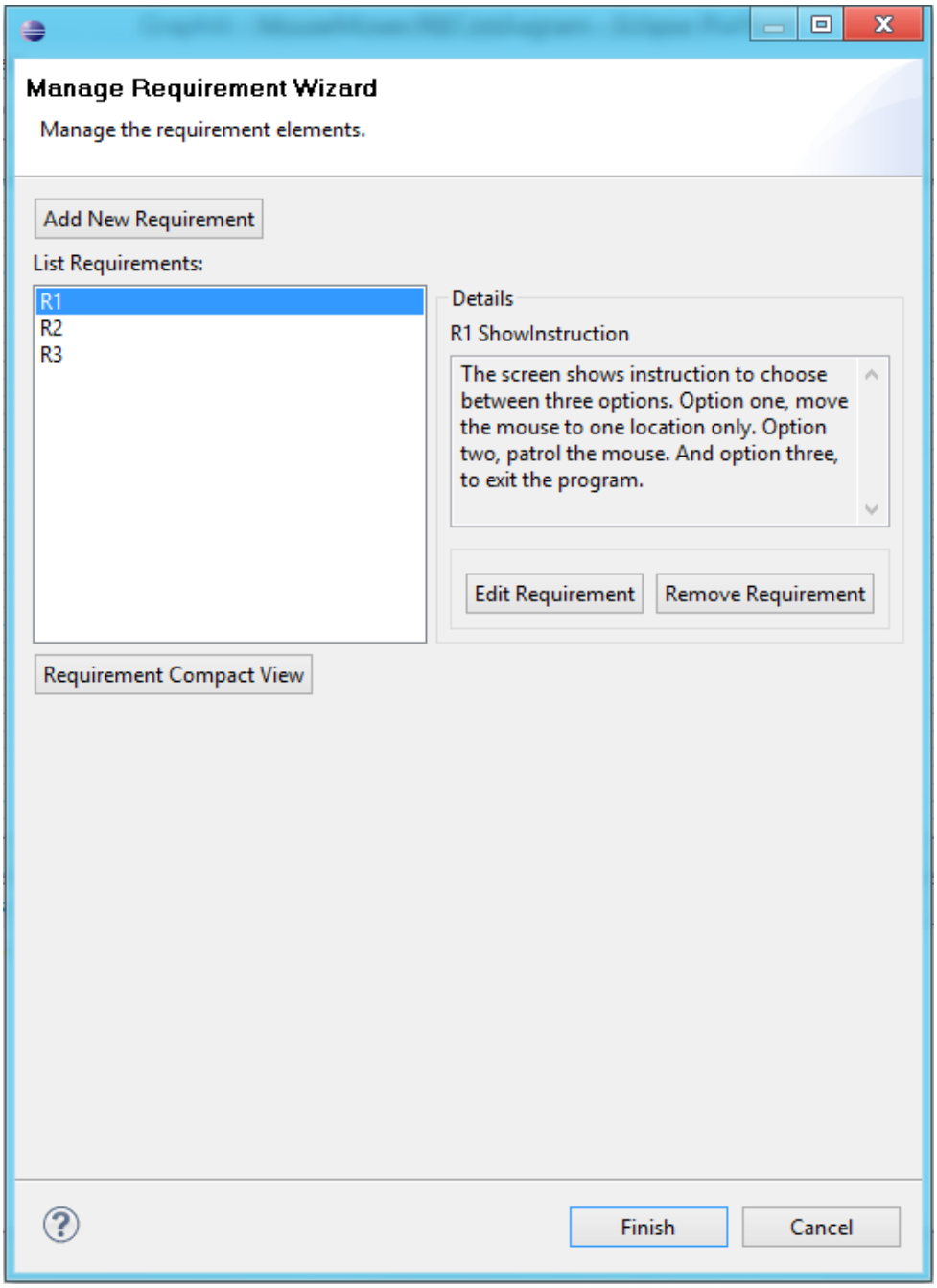


Figure 21 List of requirements for Mouse Mover

Afterwards, we can add the BT nodes which consist of two distinct Component, Host and Mouse. We can manually create the Components from the Add New Component button at the toolbar or from the Create New BT Node Wizard.

The image below will show how to create a BT node with have Host as Component and STRING_choose as Behavior and External Output as its type.

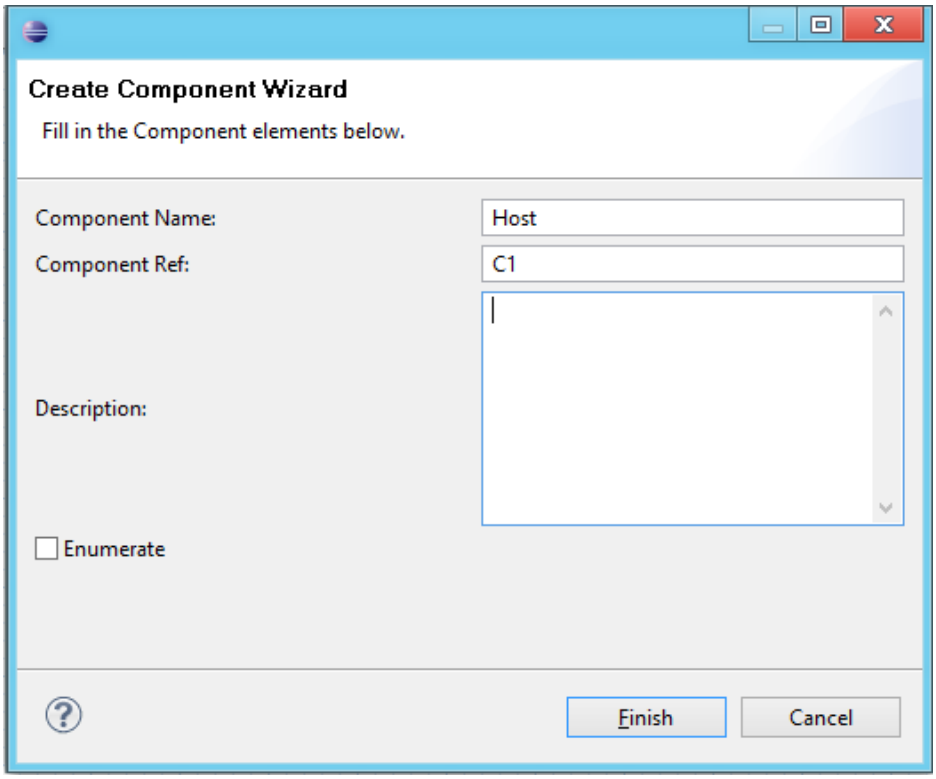


Figure 22 Creating Host as a new Component

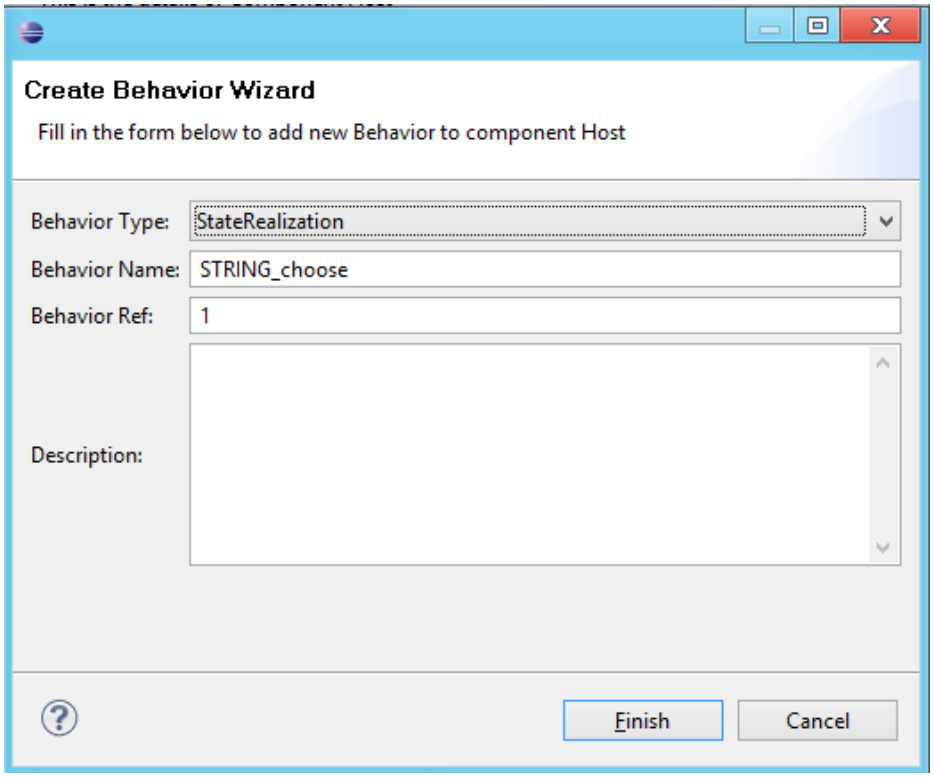


Figure 23 Creating STRING_choose as new Behavior for Host

The complete lists of Mouse Behavior is shown below

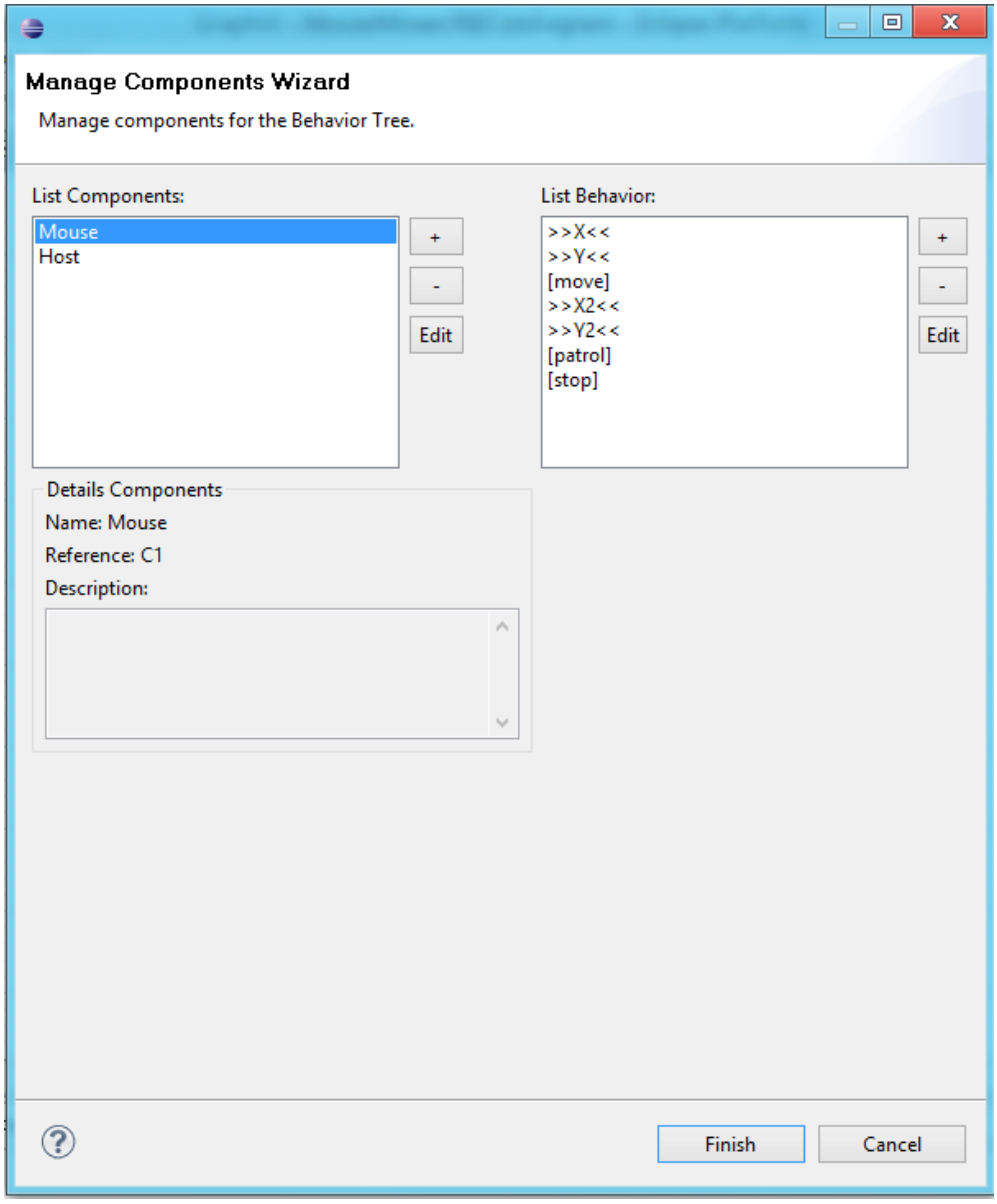


Figure 24 List of Behaviors of Mouse

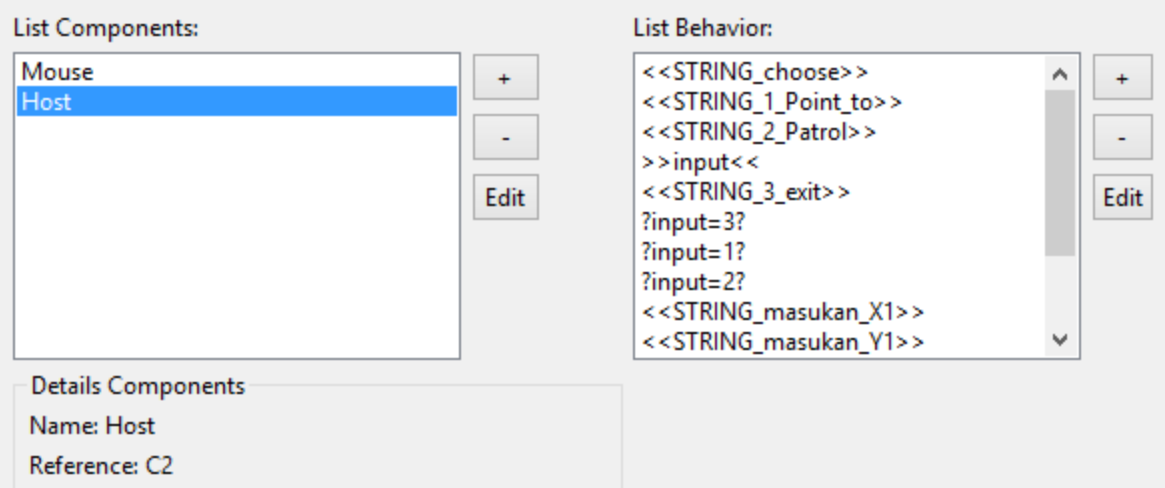


Figure 25 List of Behavior for Host (part 1 of 2)

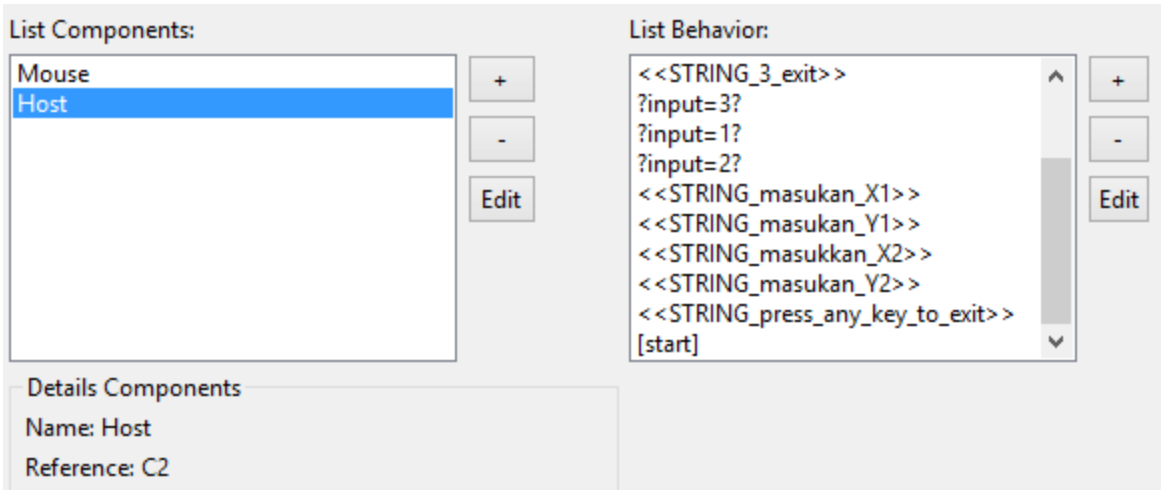


Figure 26 List of Behavior for Host (part 2 of 2)

We then add all the required nodes until the all requirements are fulfilled. The final diagram of this example is shown below.

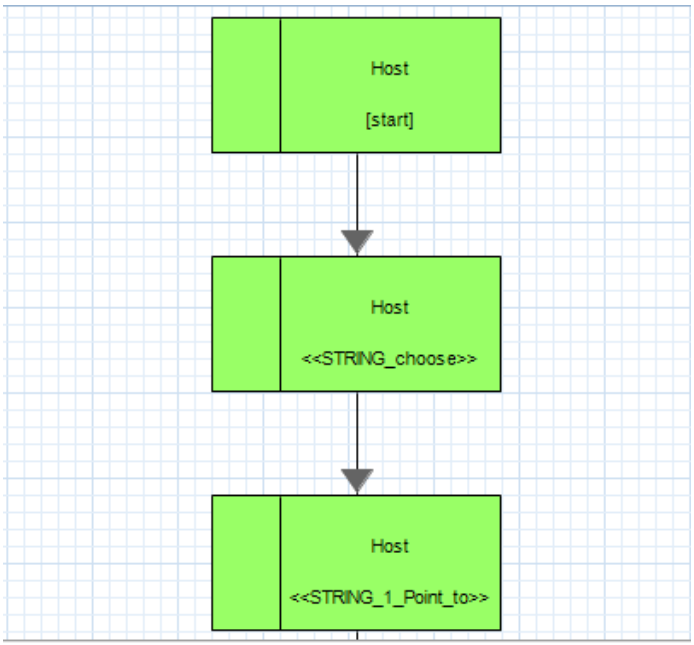


Figure 27 Complete Behavior Tree diagram for Mouse Mover (part 1 of 6)

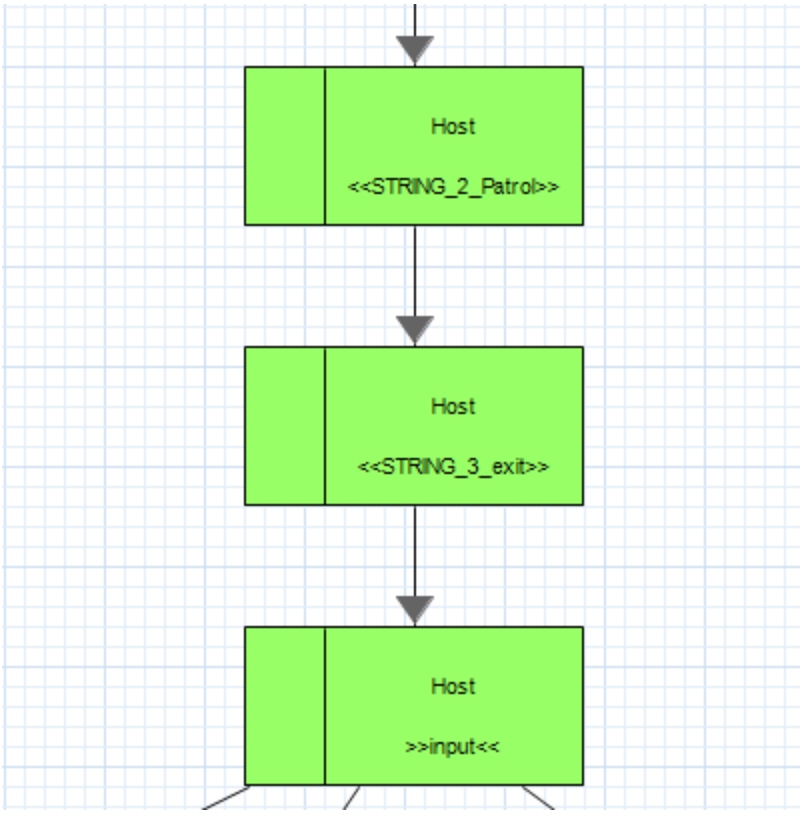


Figure 28 Complete Behavior Tree diagram for Mouse Mover (part 2 of 6)

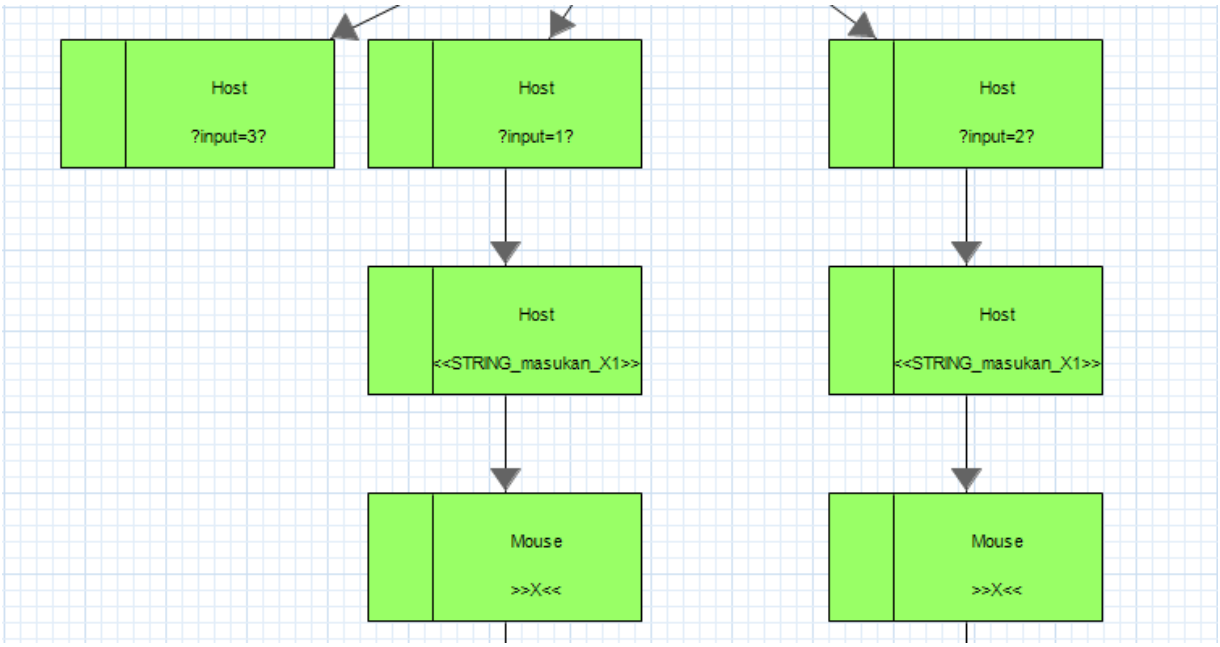


Figure 29 Complete Behavior Tree diagram for Mouse Mover (part 3 of 6)

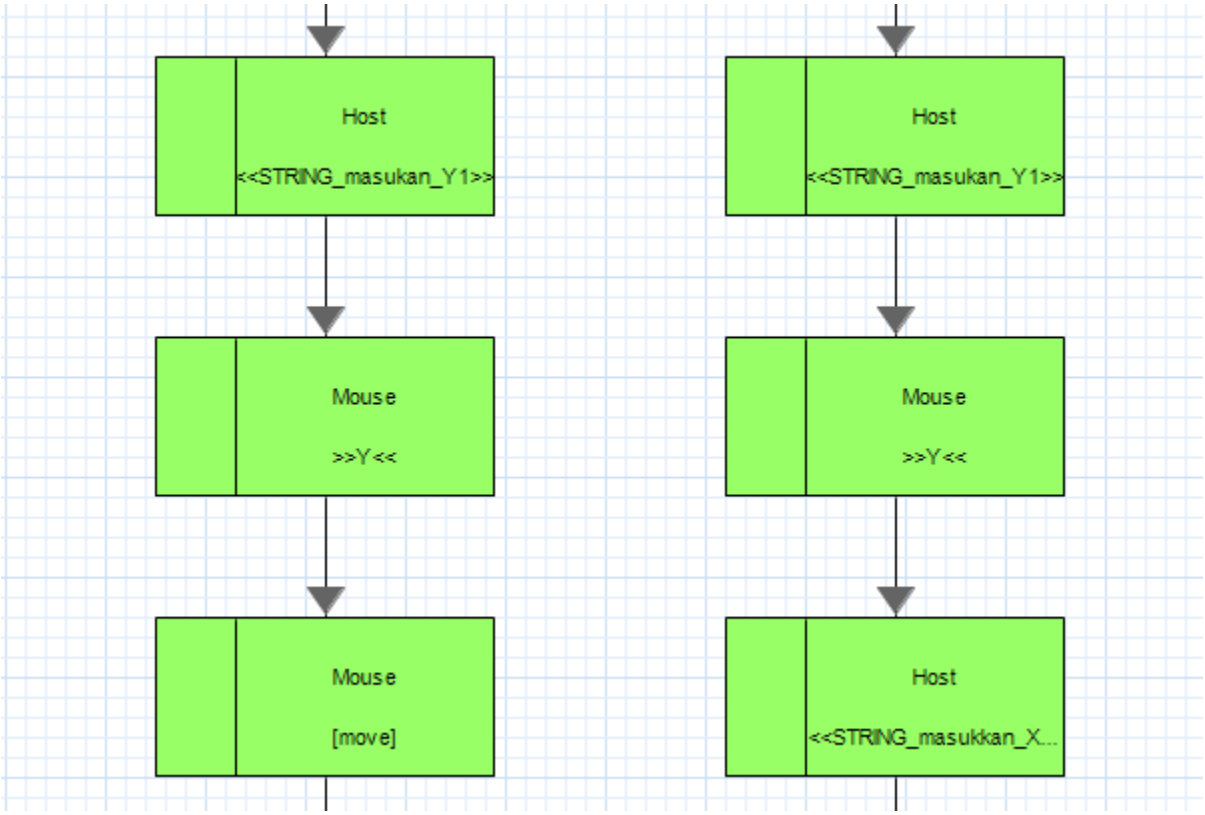


Figure 30 Complete Behavior Tree diagram for Mouse Mover (part 4 of 6)

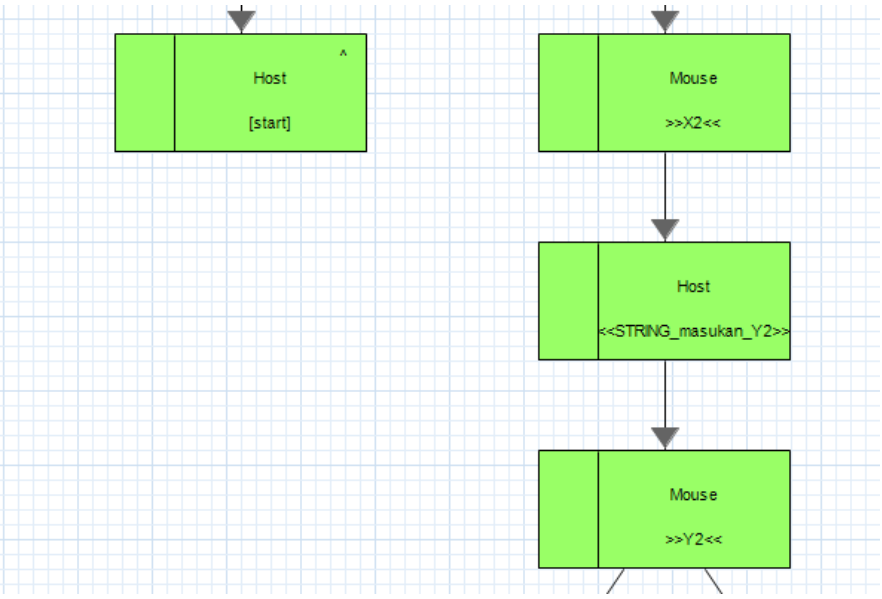


Figure 31 Complete Behavior Tree diagram for Mouse Mover (part 5 of 6)

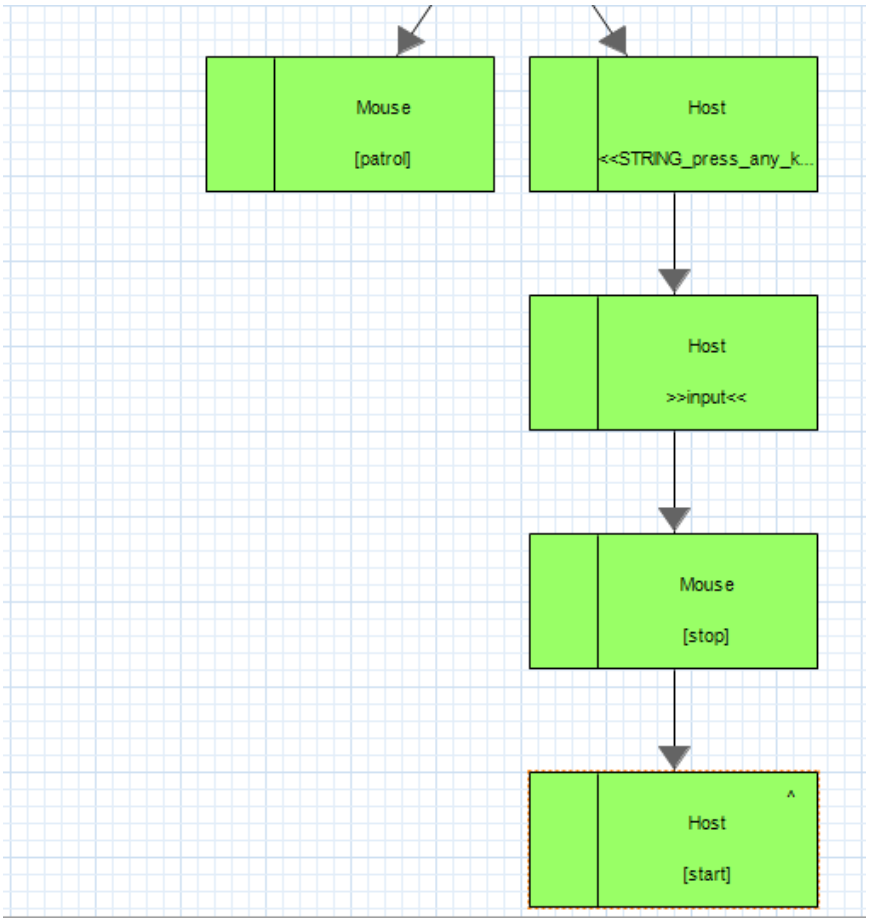


Figure 32 Complete Behavior Tree diagram for Mouse Mover (part 6 of 6)