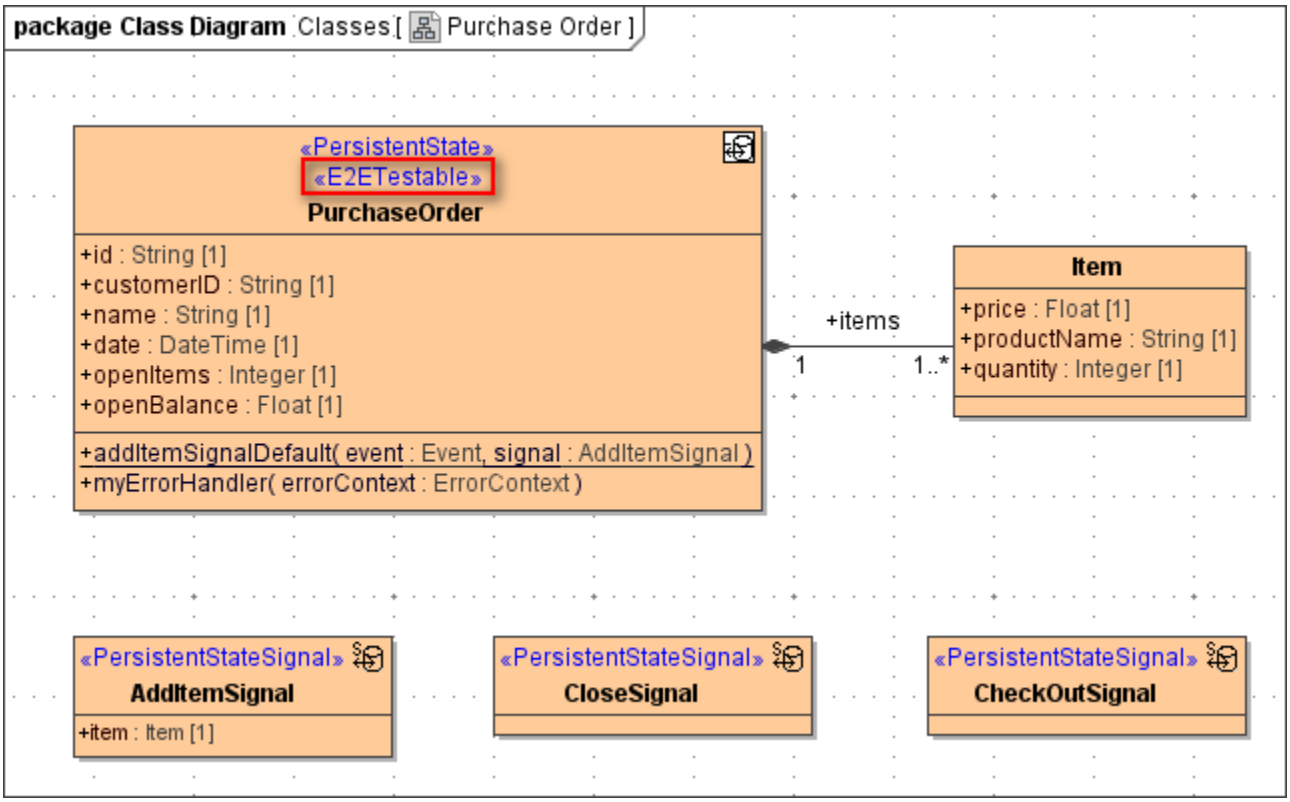
Testable Classes and Persistent States

The usage of the stereotype <<E2ETestable>> is allowed on persistent state classes as described in the topics before.

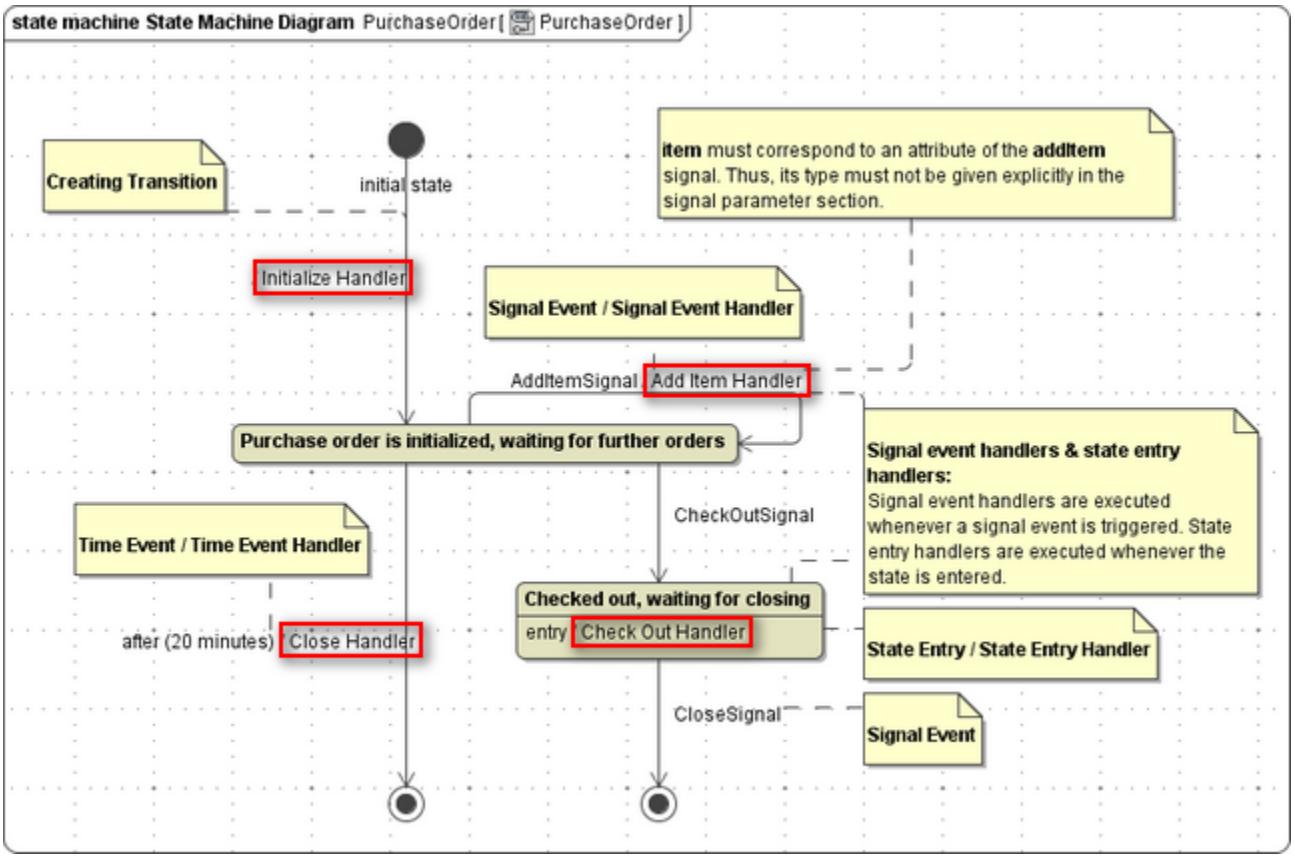
Figure: Persistent State Class with Stereotype <<E2ETestable>>



Making persistent state classes testable gives the modeler the opportunity to test the activities implemented on the state transitions independently - without bothering about the persistent state database.

> Testing a persistent state class by the use of <<E2ETestable>> will not affect the persistent state database.

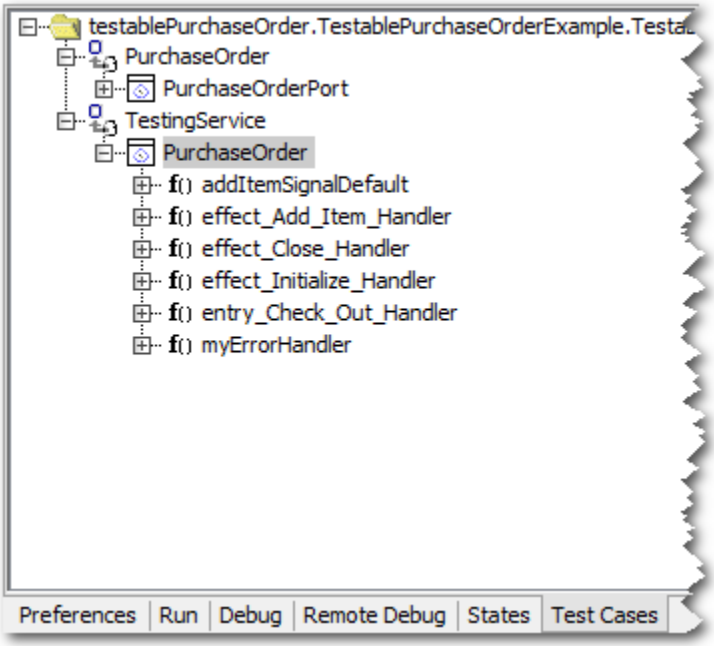
Figure: State Machine Diagram of Purchase Order



The E2E Builder generates test operations to test all activities related to the persistent state class, which are

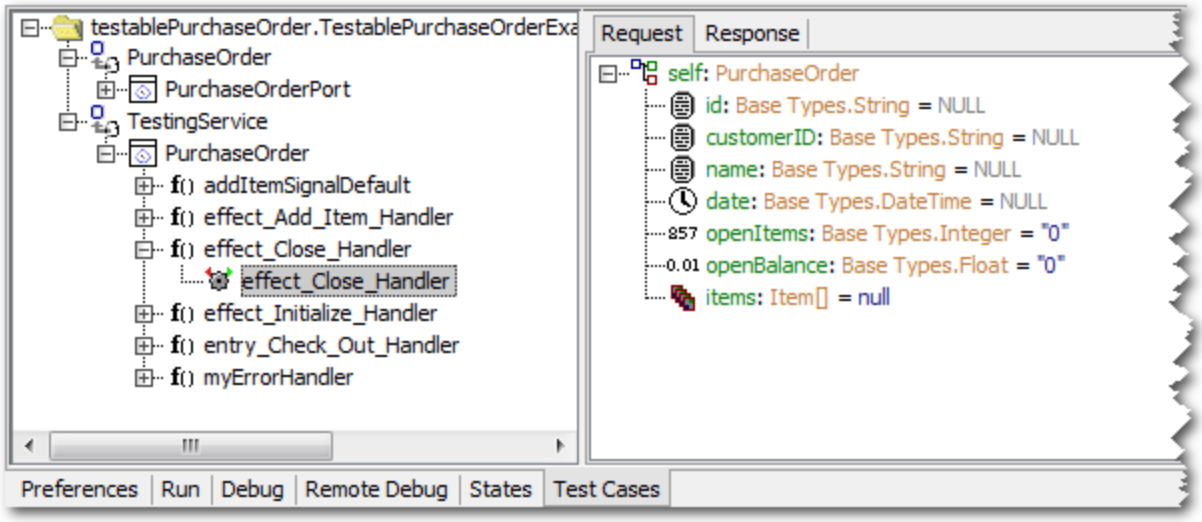
- operations defined on the class itself (e.g. **addItemSignalDefault** and **myErrorHandler**)
- activities related to transitions (e.g. **Add Item Handler**, **Close Handler** and **Initialize Handler**, marked in red in the figure above)
- entry state and exit state activities (e.g. **Check Out Handler**, marked in red in the figure above)
- do activities

Figure: Generated Testing Interface for Persistent State Class PuchaseOrder



The test interface for a normal signal transition will present as shown below. All persistent state related test cases will get the self context of the persistent state class in order to enter data relevant for testing.

Figure: Transition Test Case



The table below shows an overview of all activities, their generated default names within the testing interface and the request parameters offered by the related test cases.

Persistent State Activity	Generated Default Name	Example Reference	Request Parameters
class operation	<name of the operation>	addItemSignalDefault	• parameters specified on the operation
transition / signal	effect_<name of the activity>	effect_Add_Item_Handler	• self • additional properties specified on the signal class
entry and exit state	entry_<name of the activity> exit_<name of the activity>	entry_Check_Out_Handler	• self
do activity	doActivity_<name of the activity>		• self