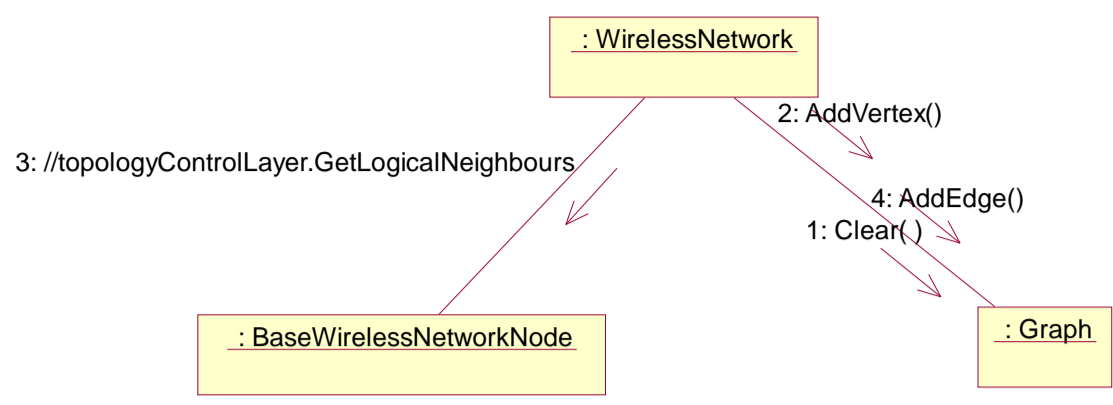
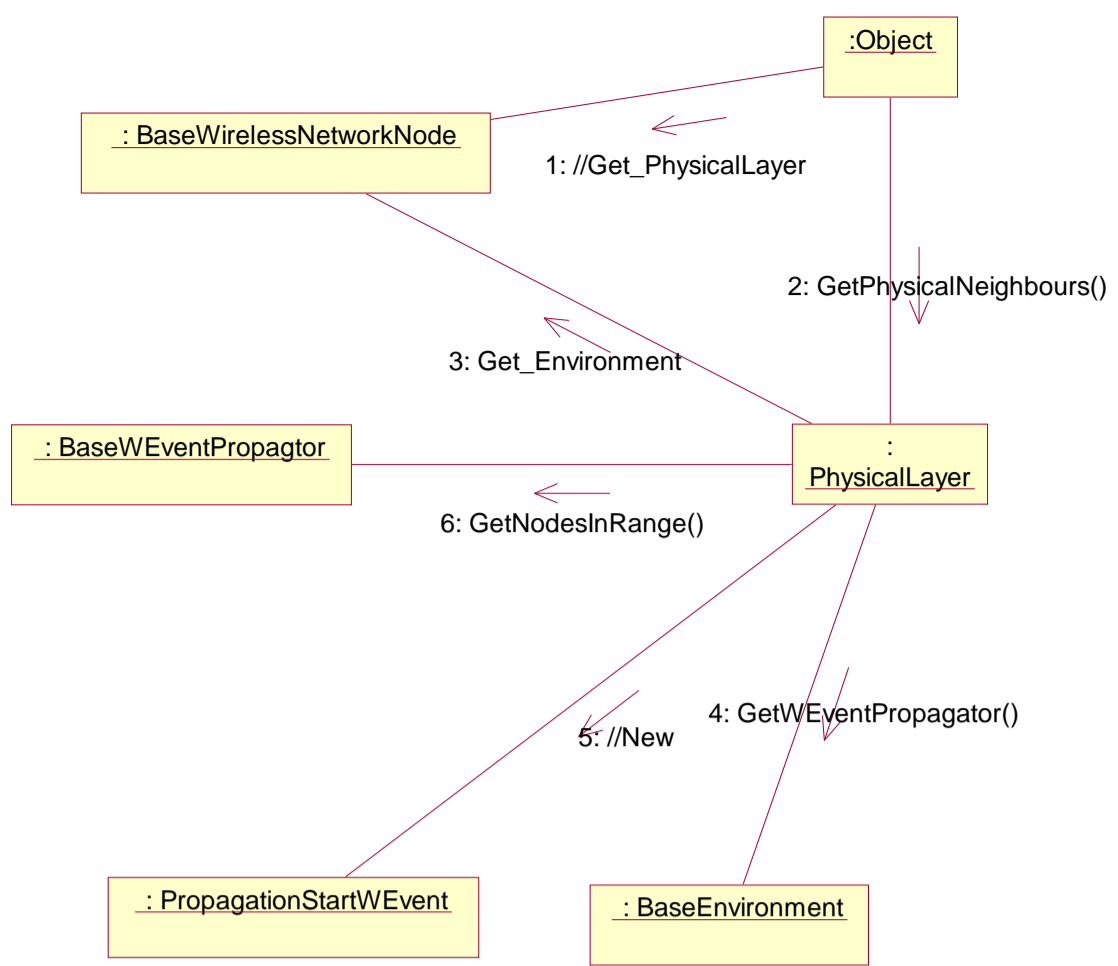
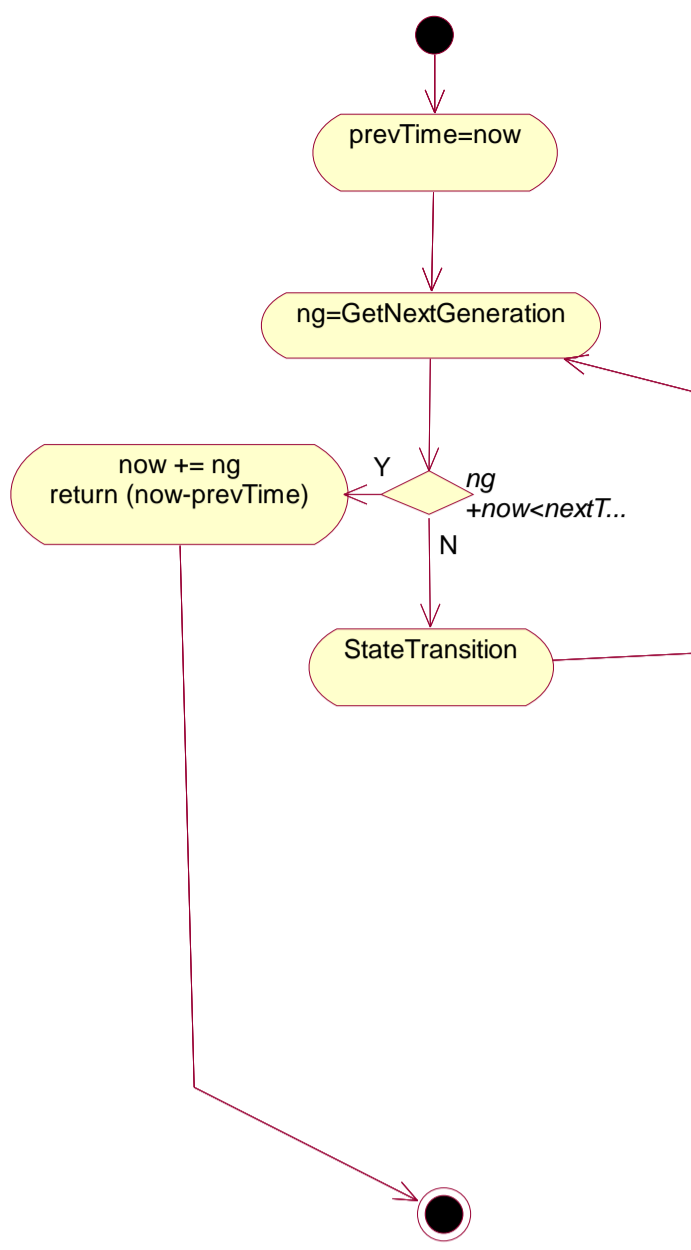


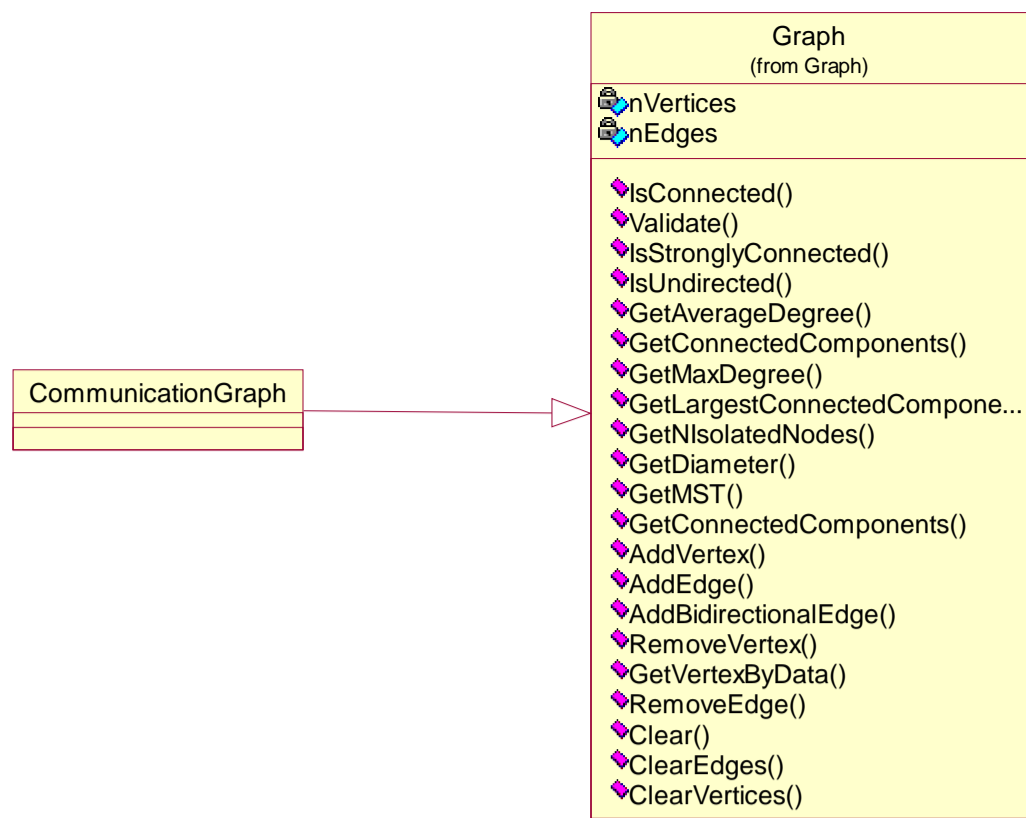
3,4 are repeated

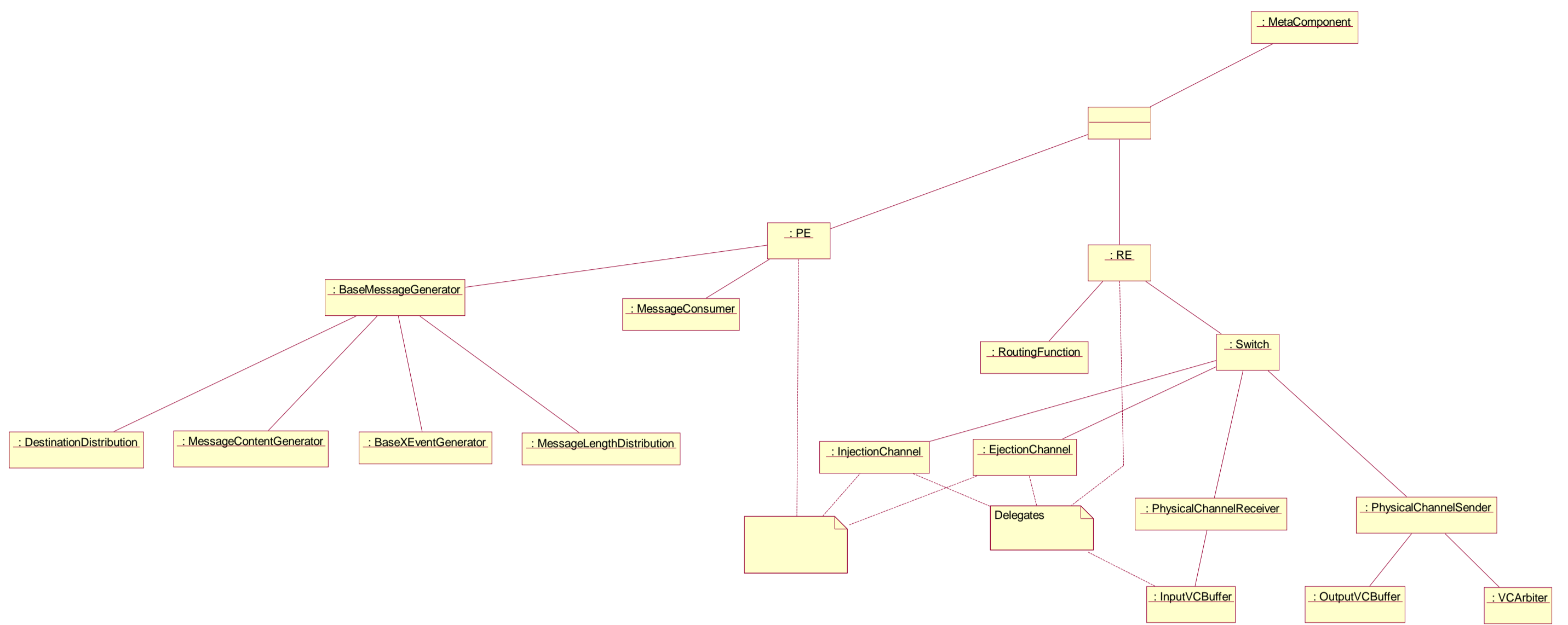


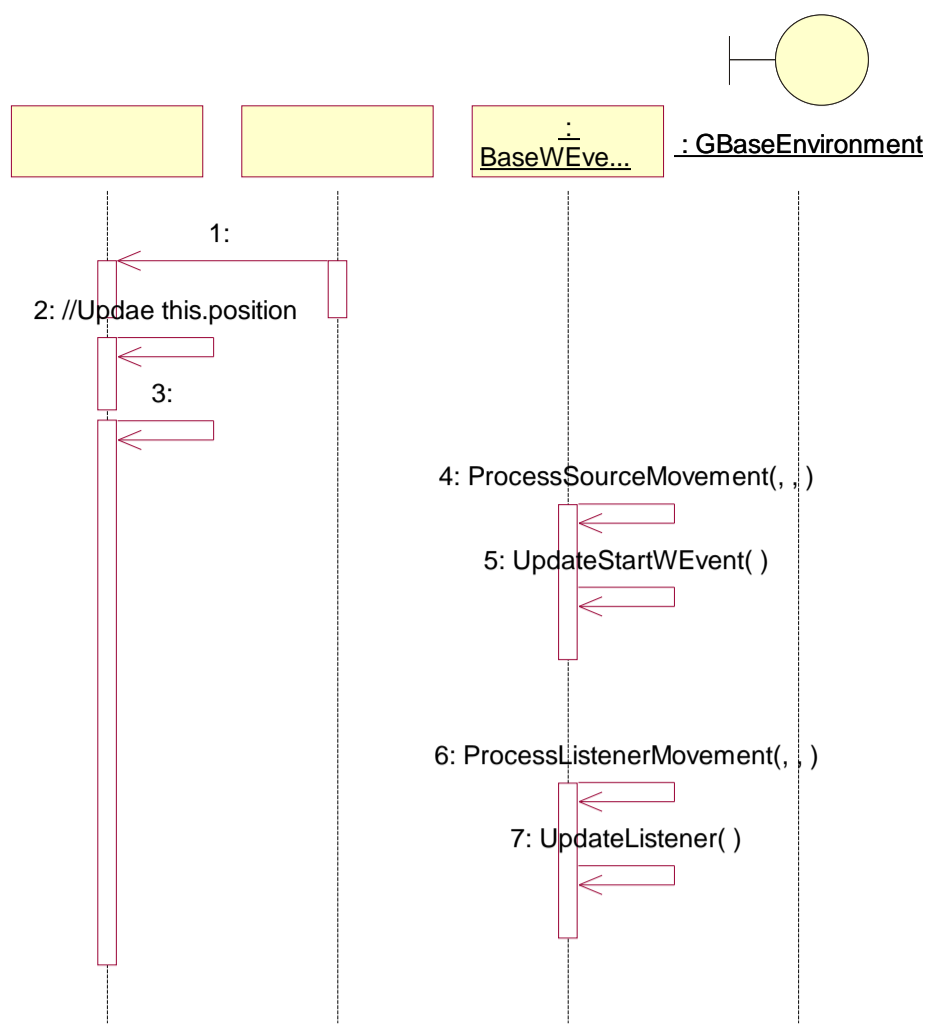
3,4 are repeated

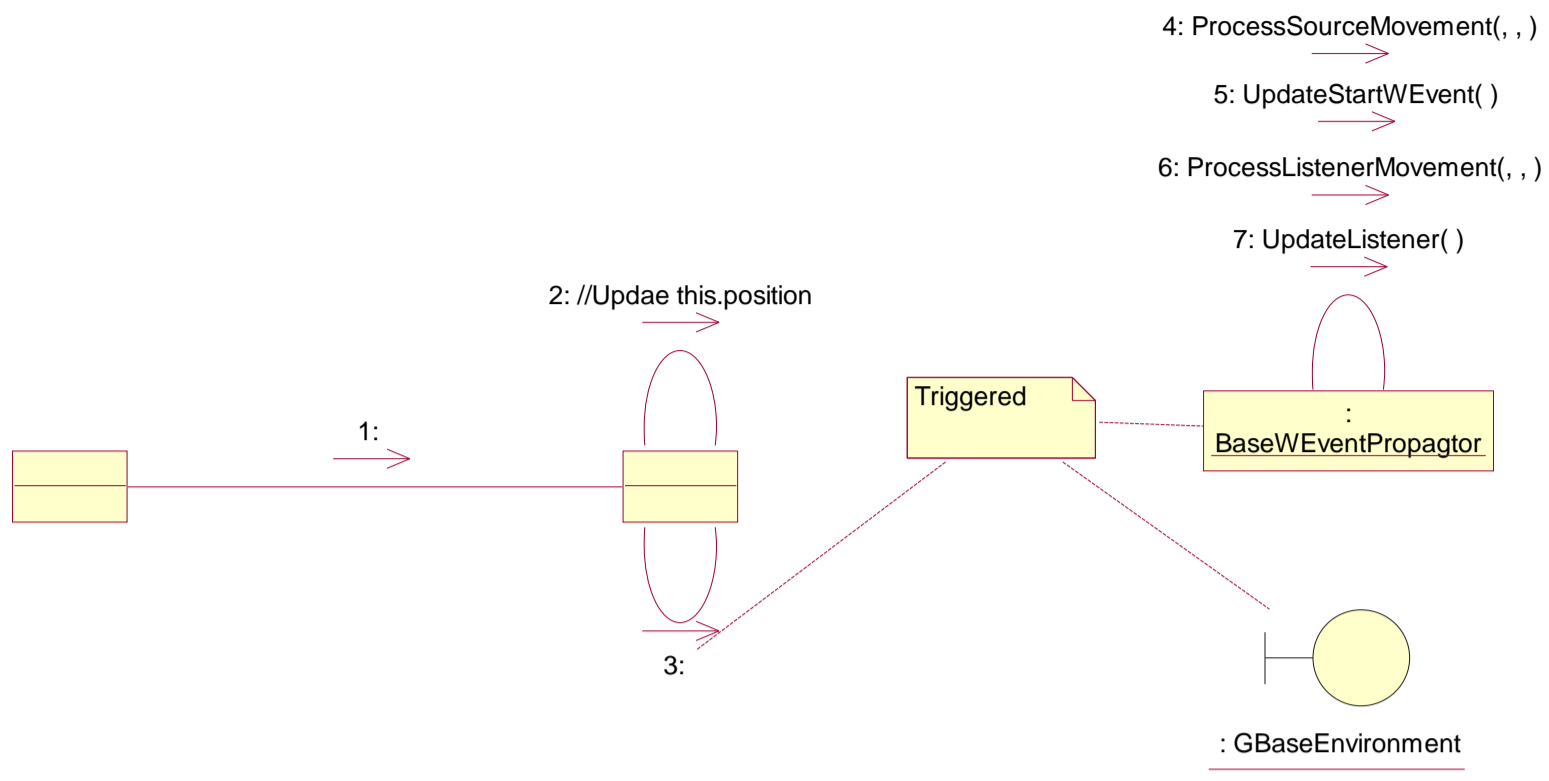




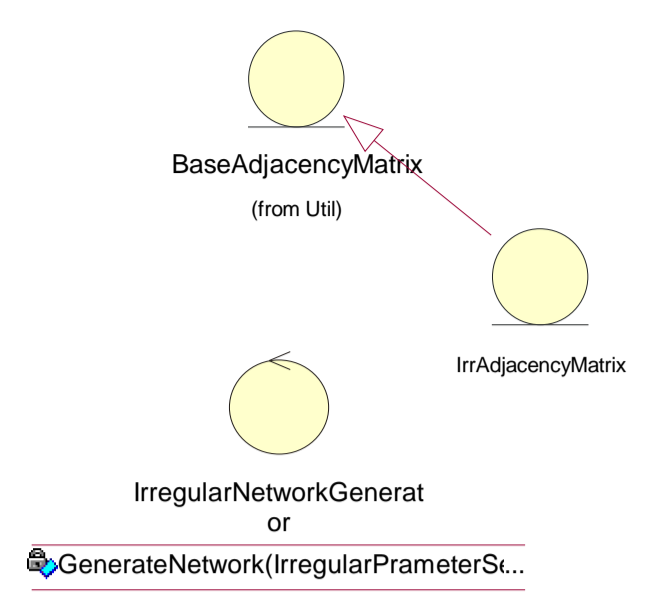
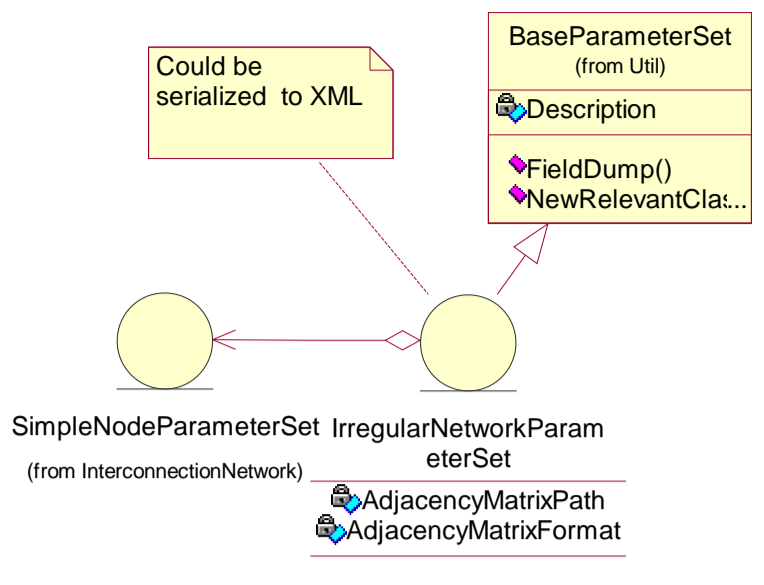


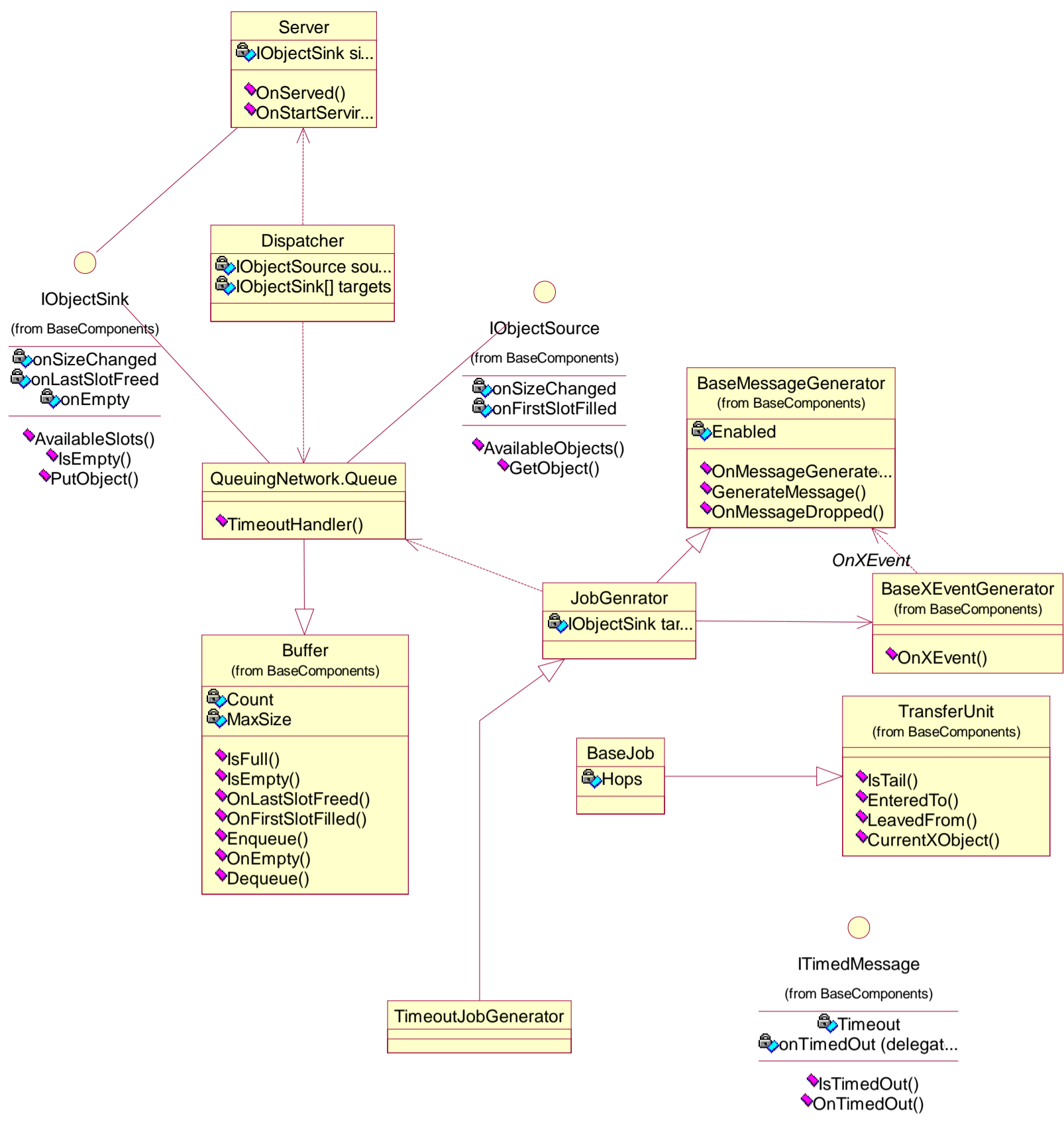


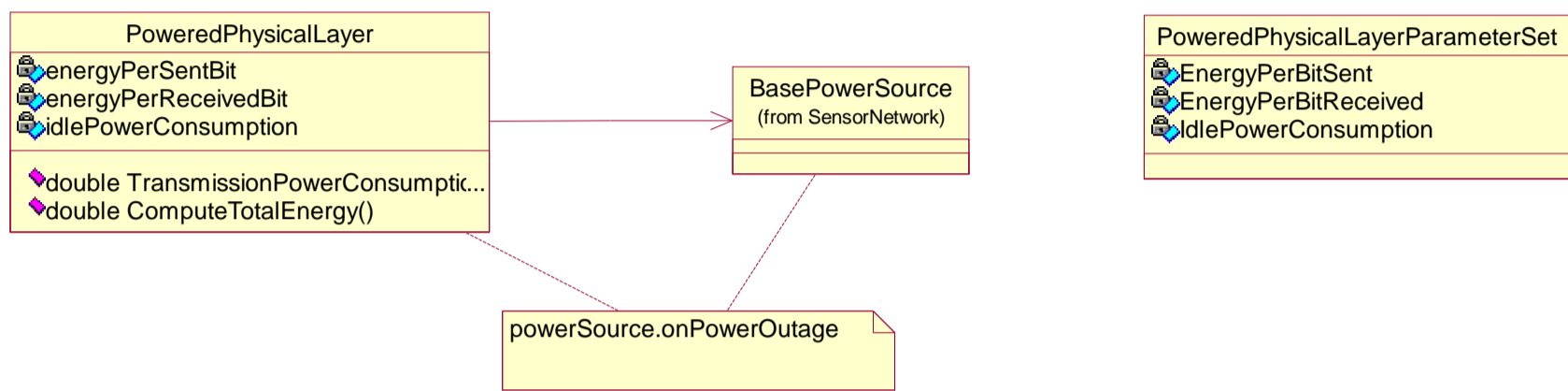
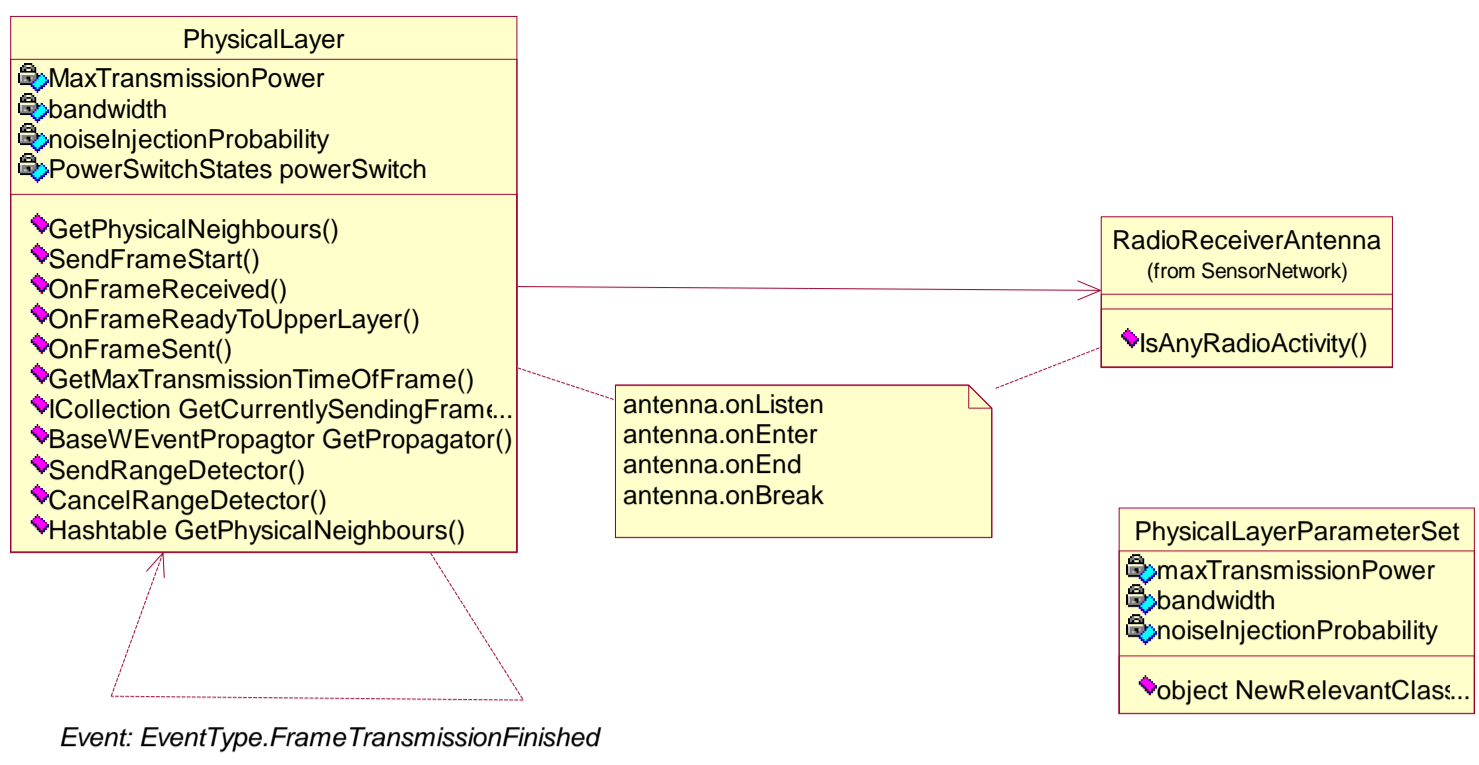


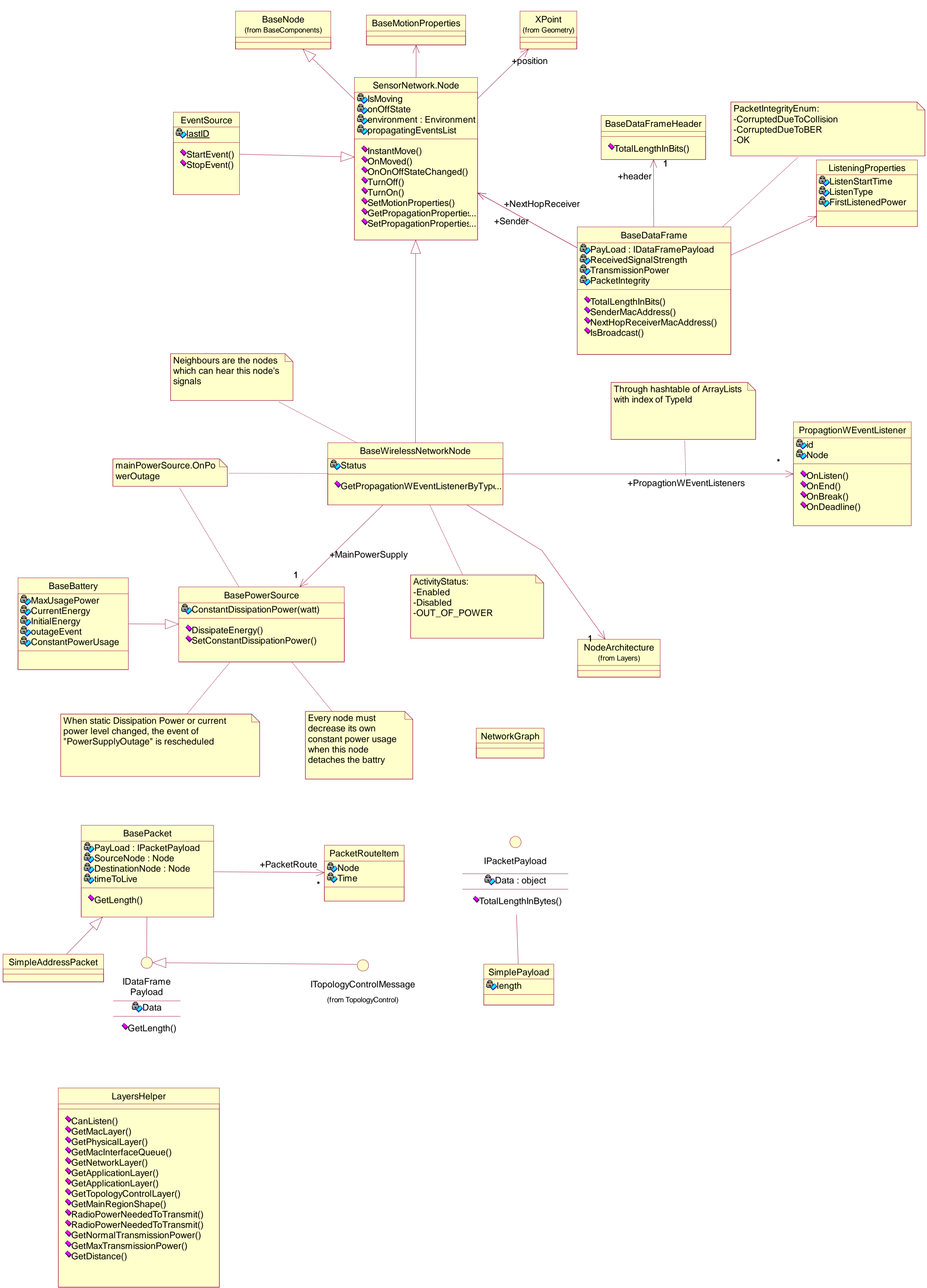


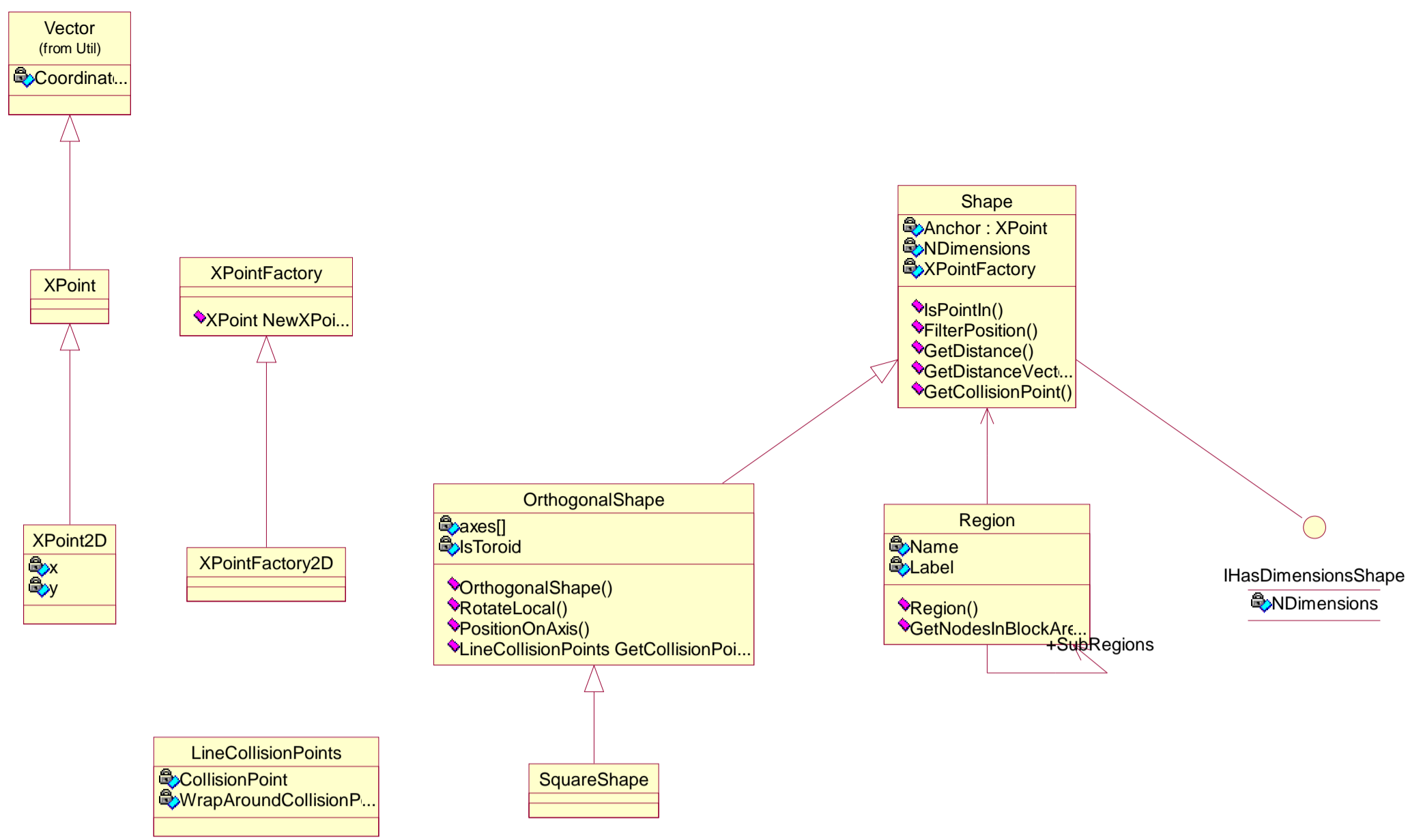


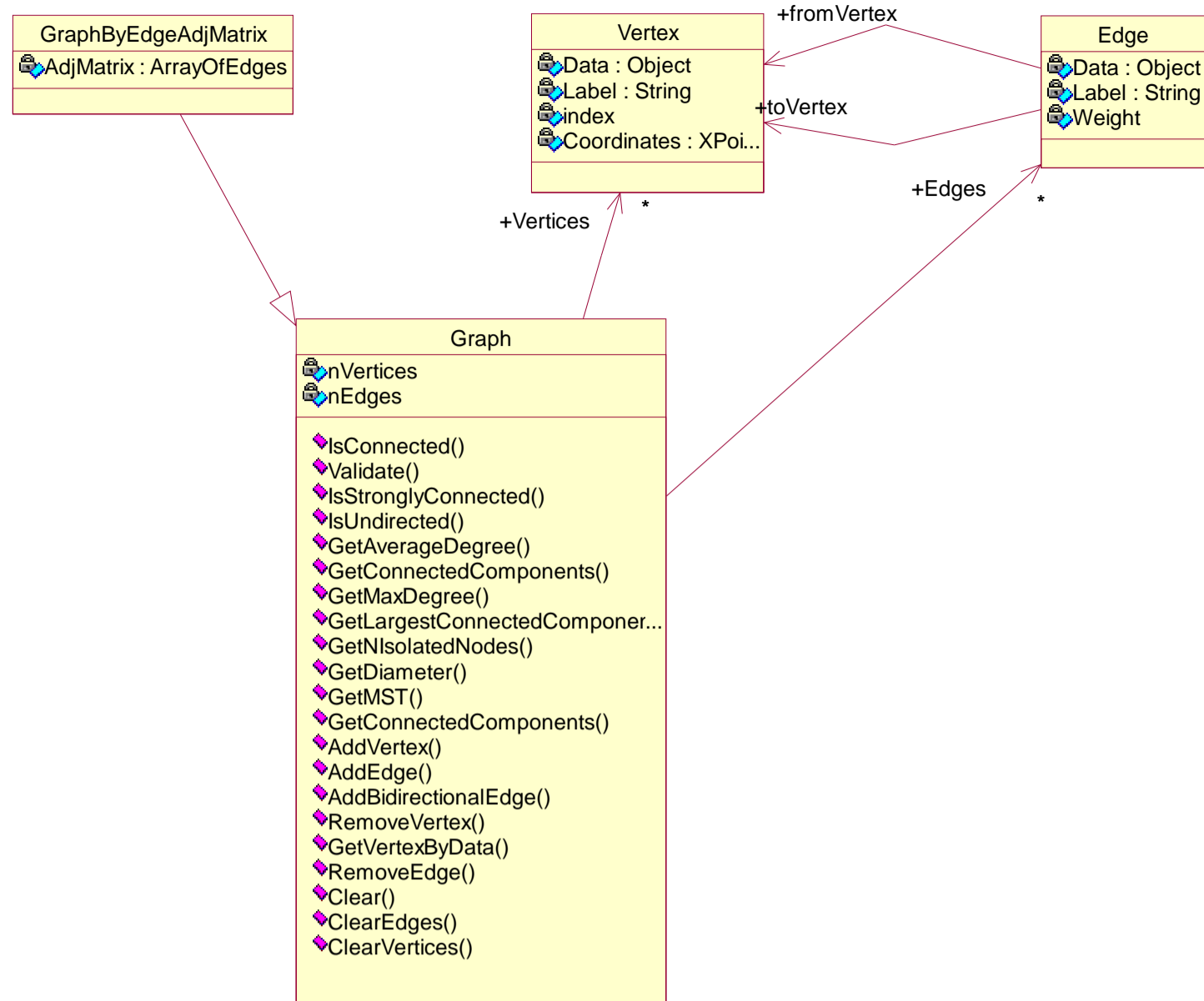


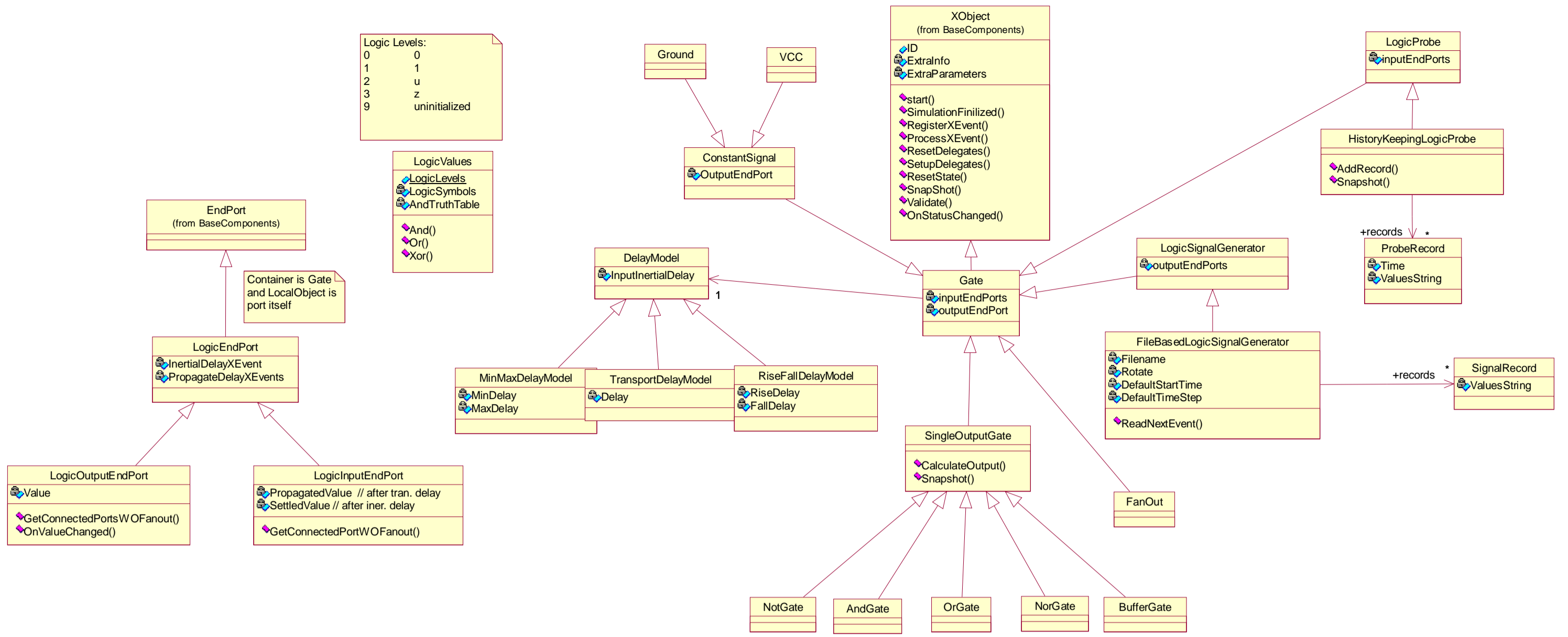


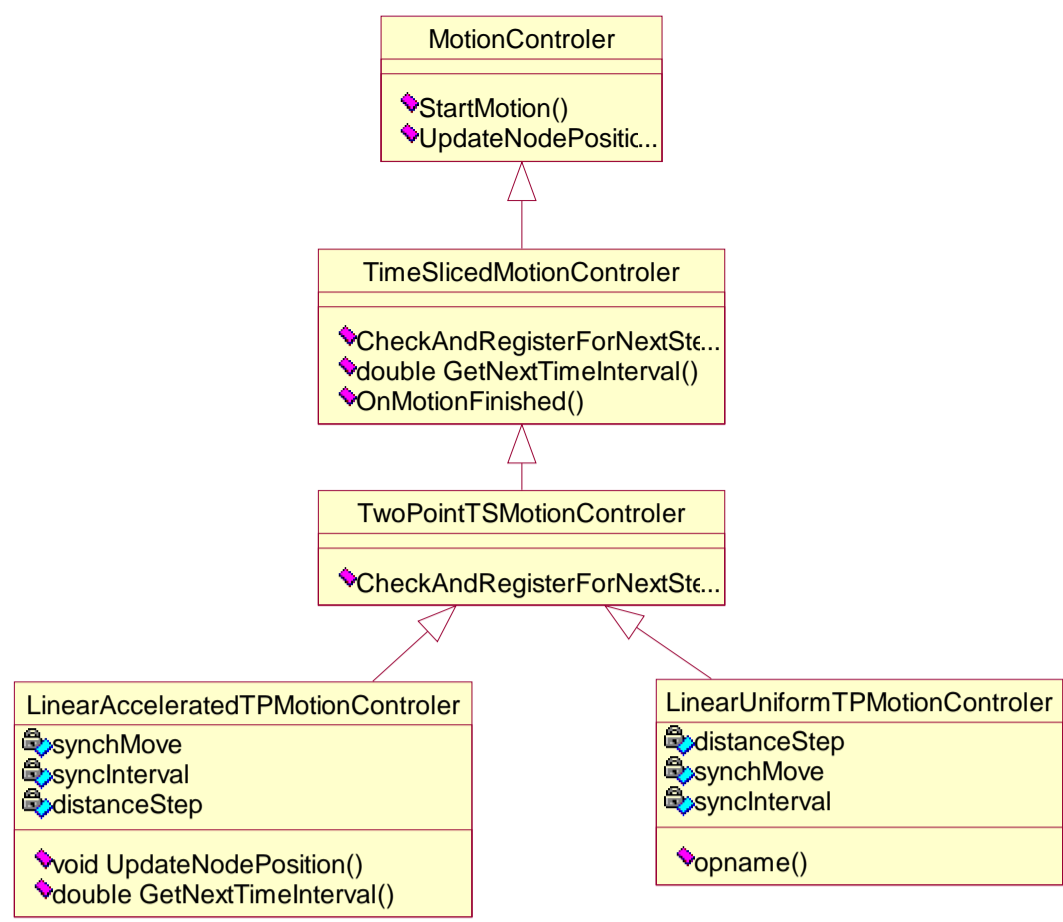




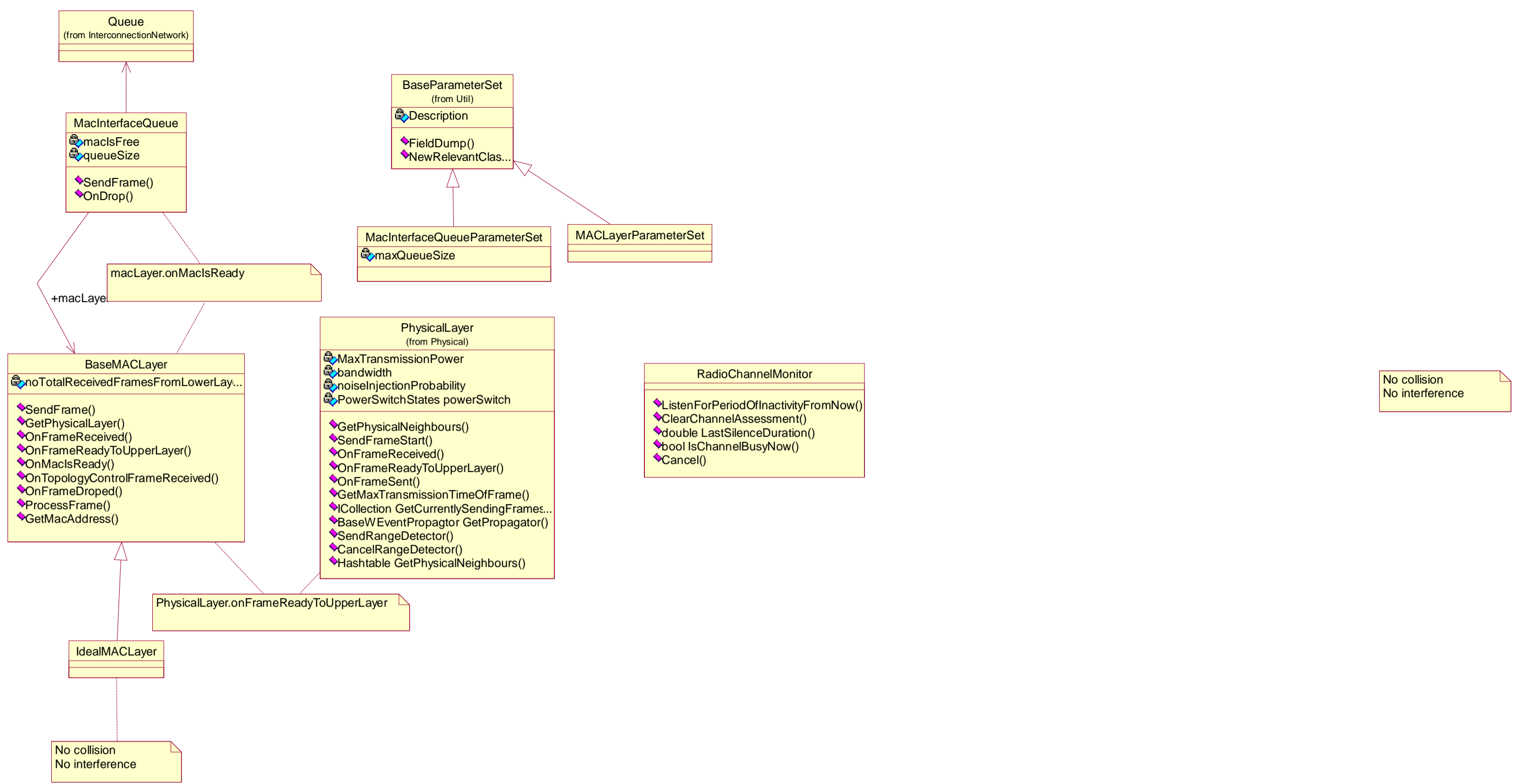


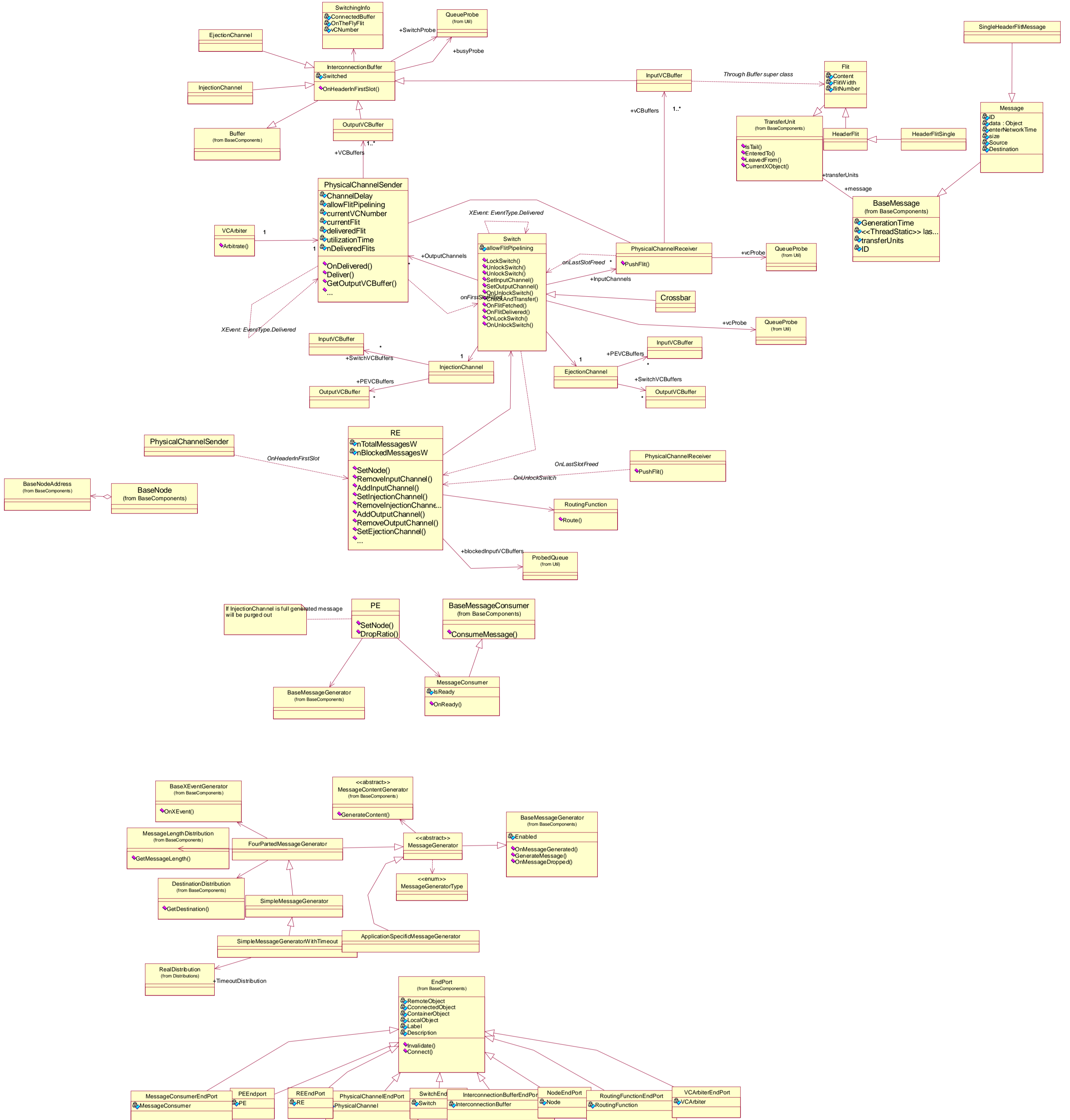


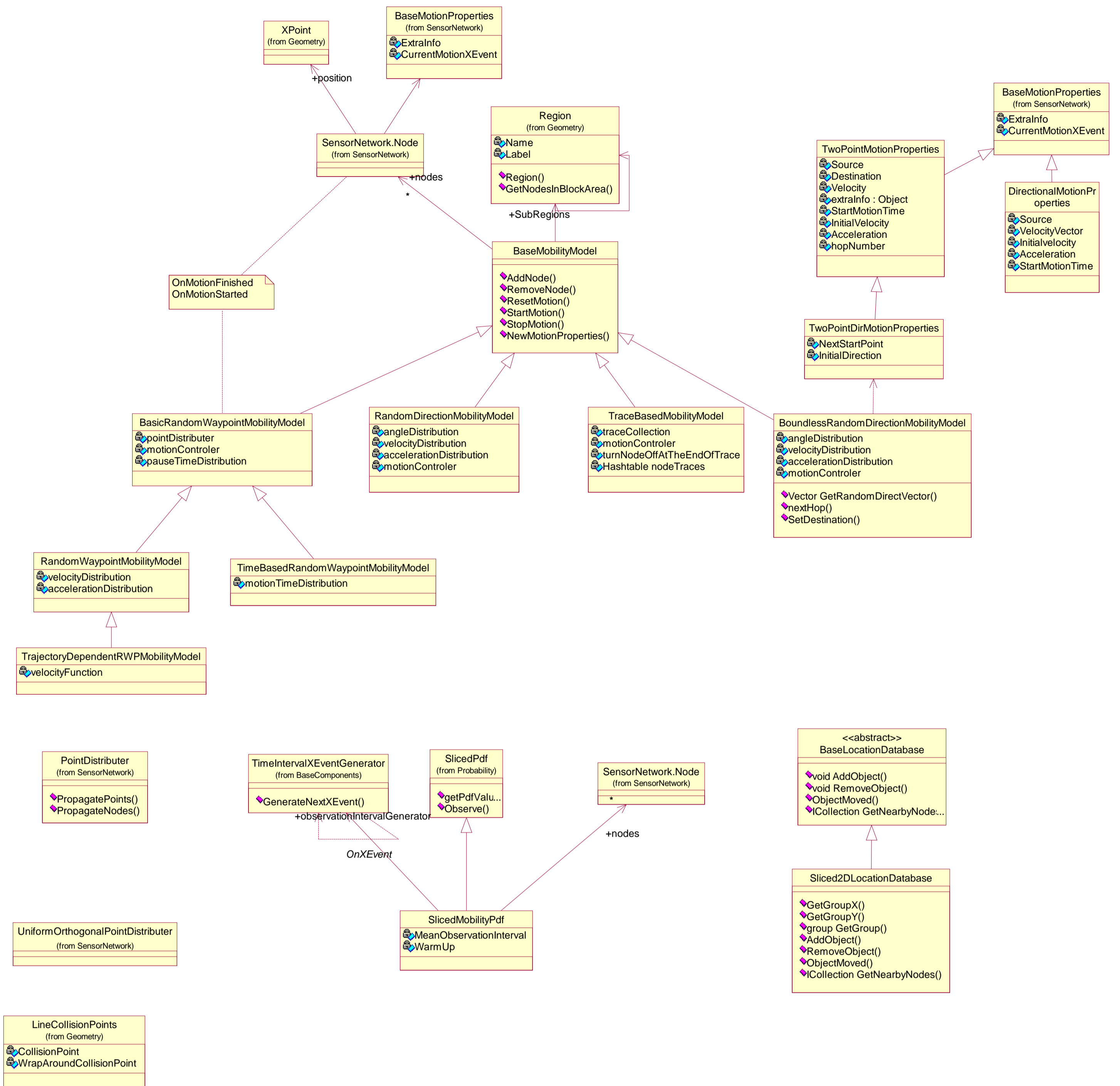


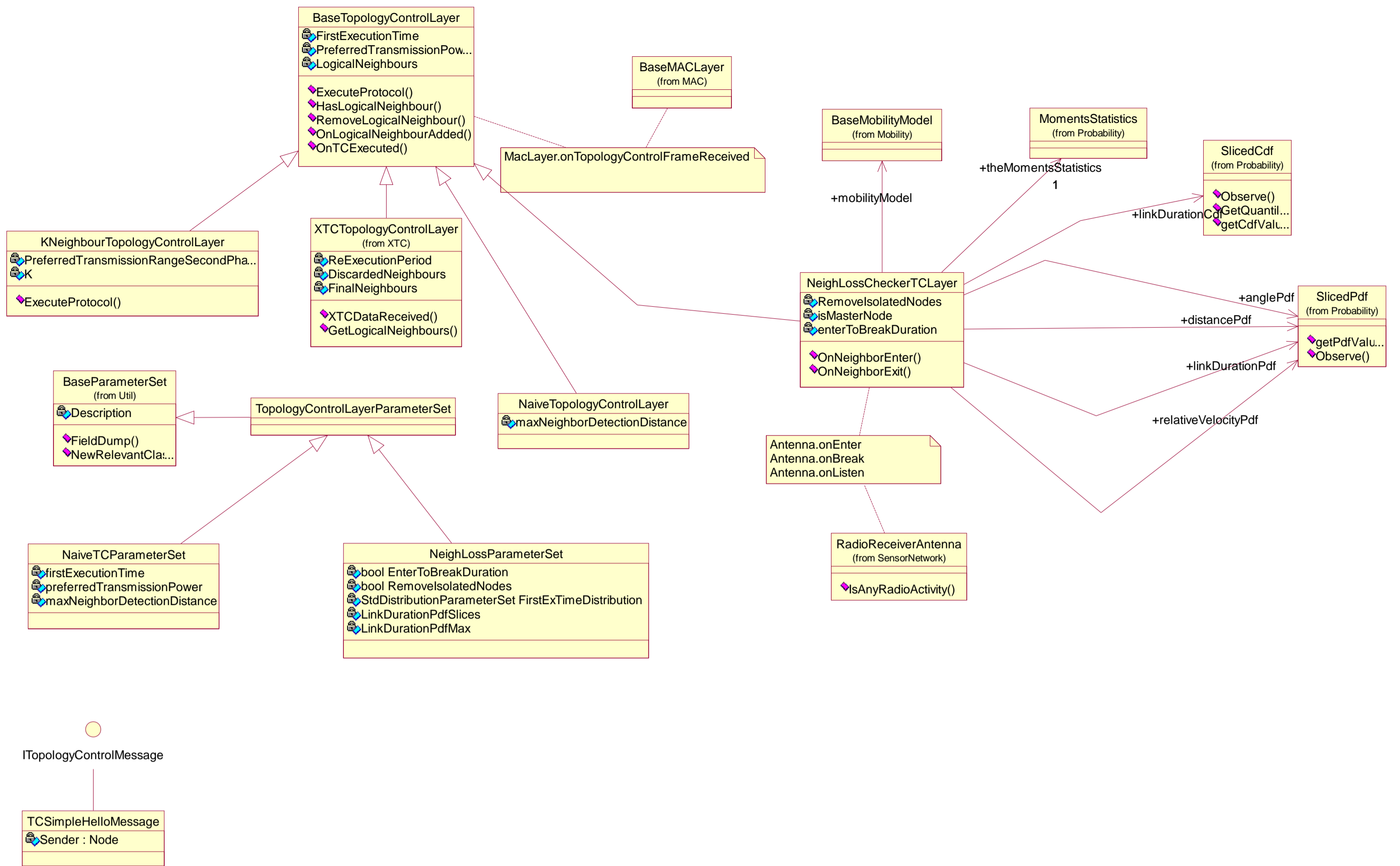


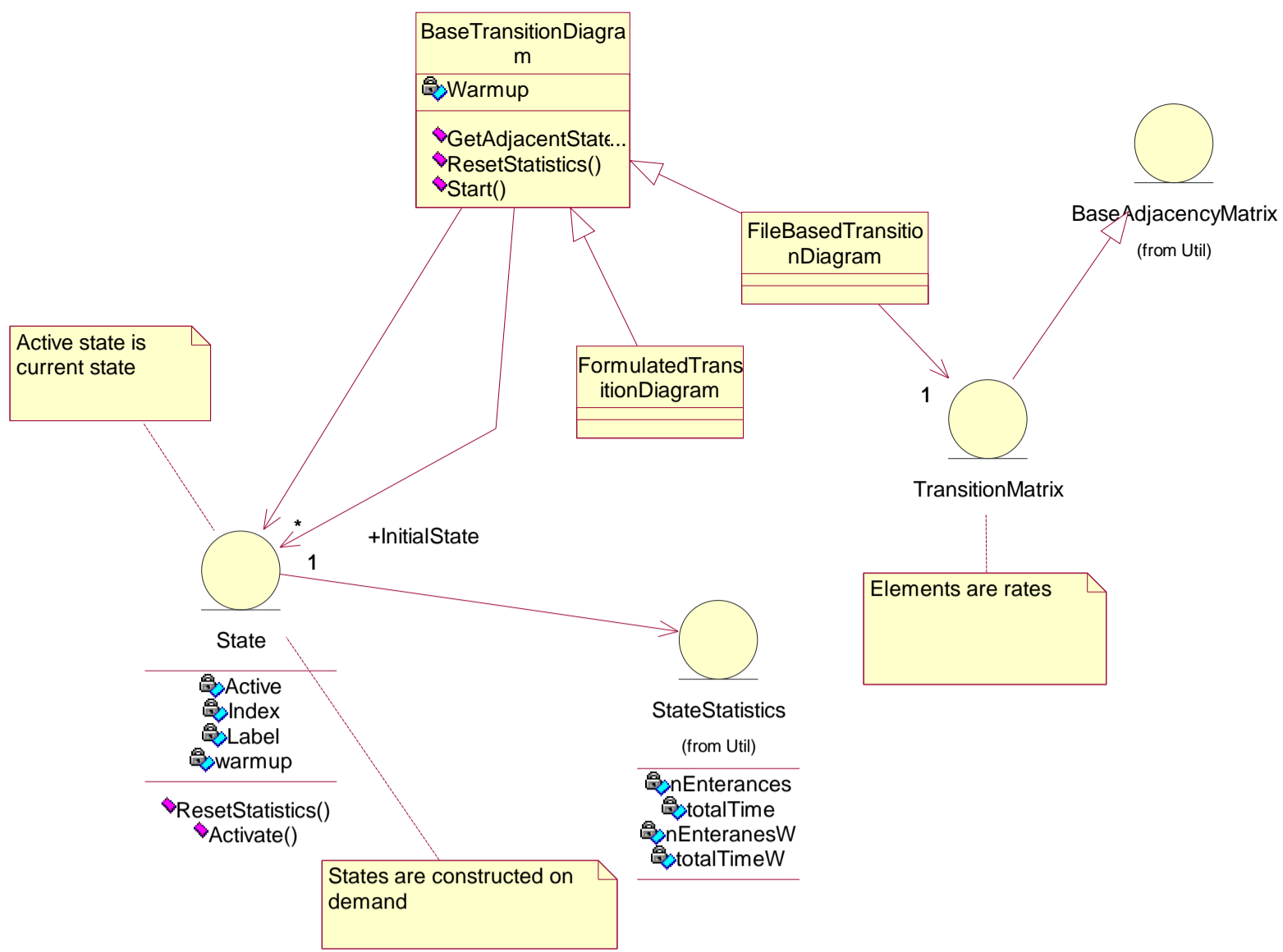


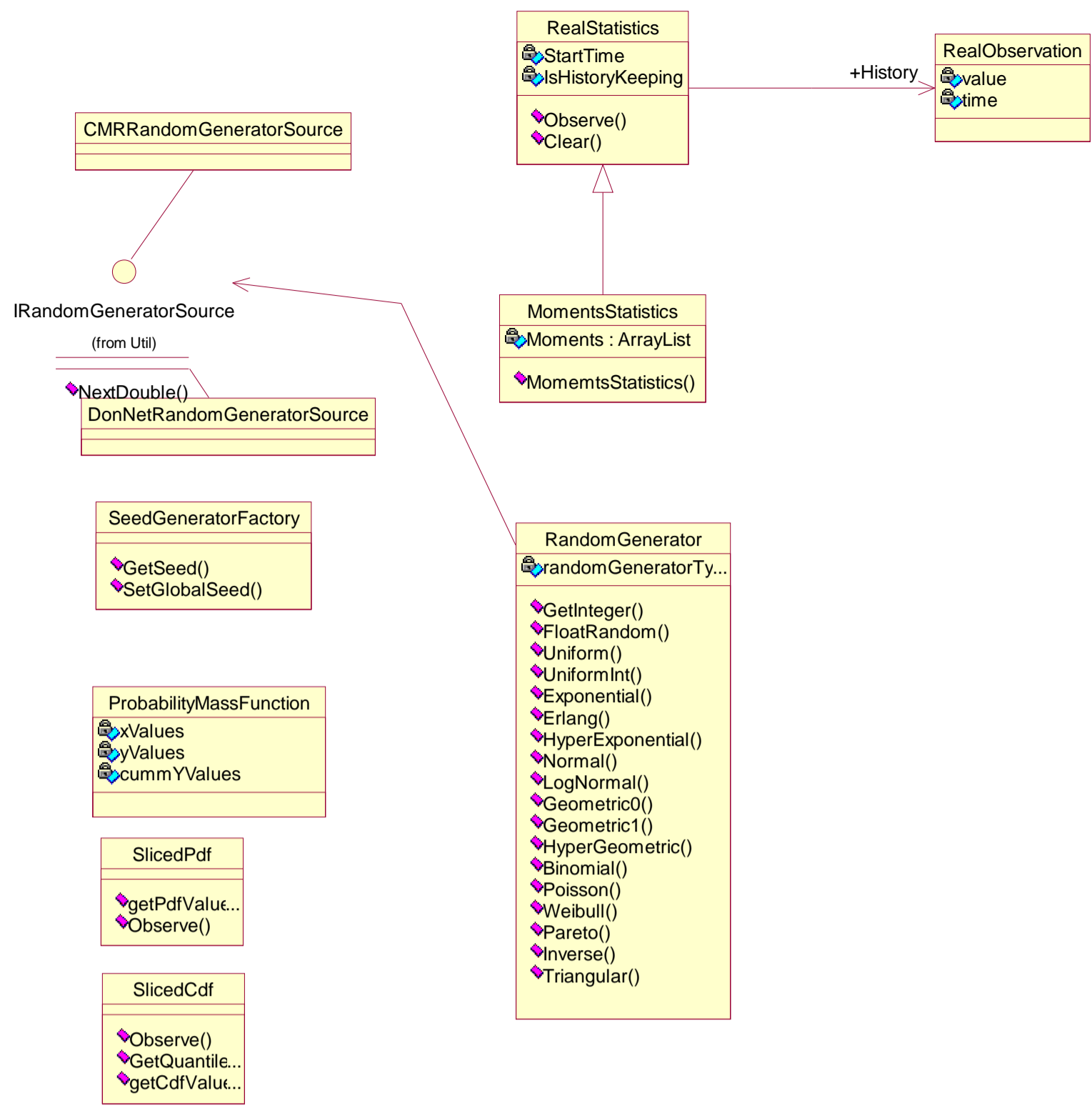




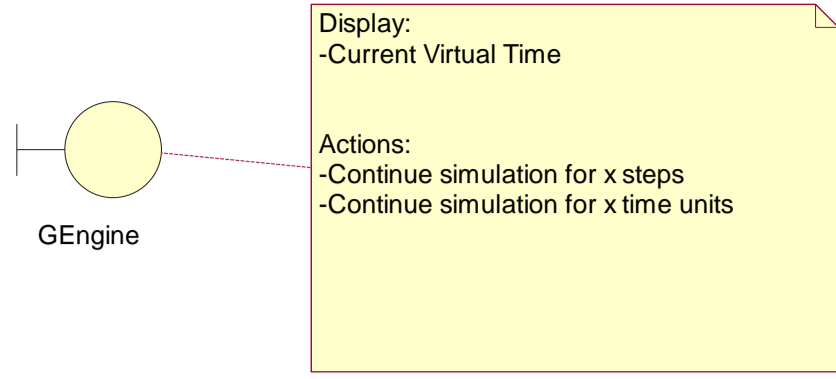




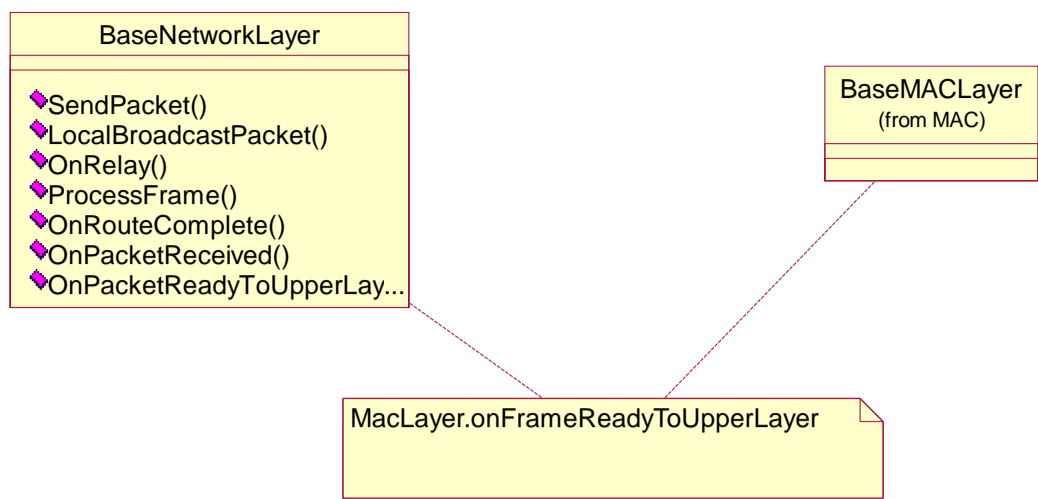


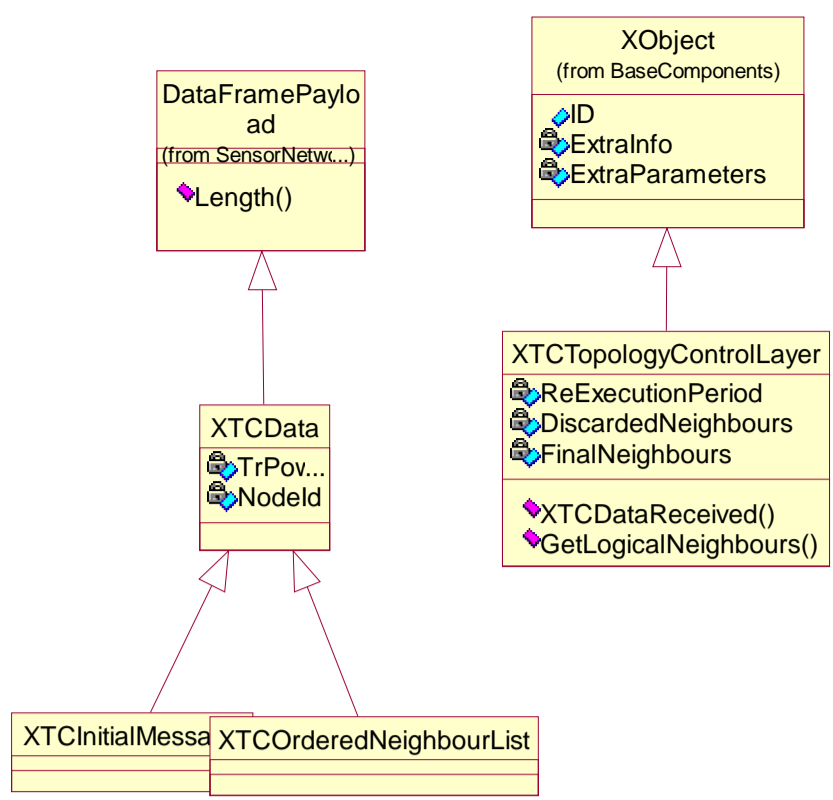


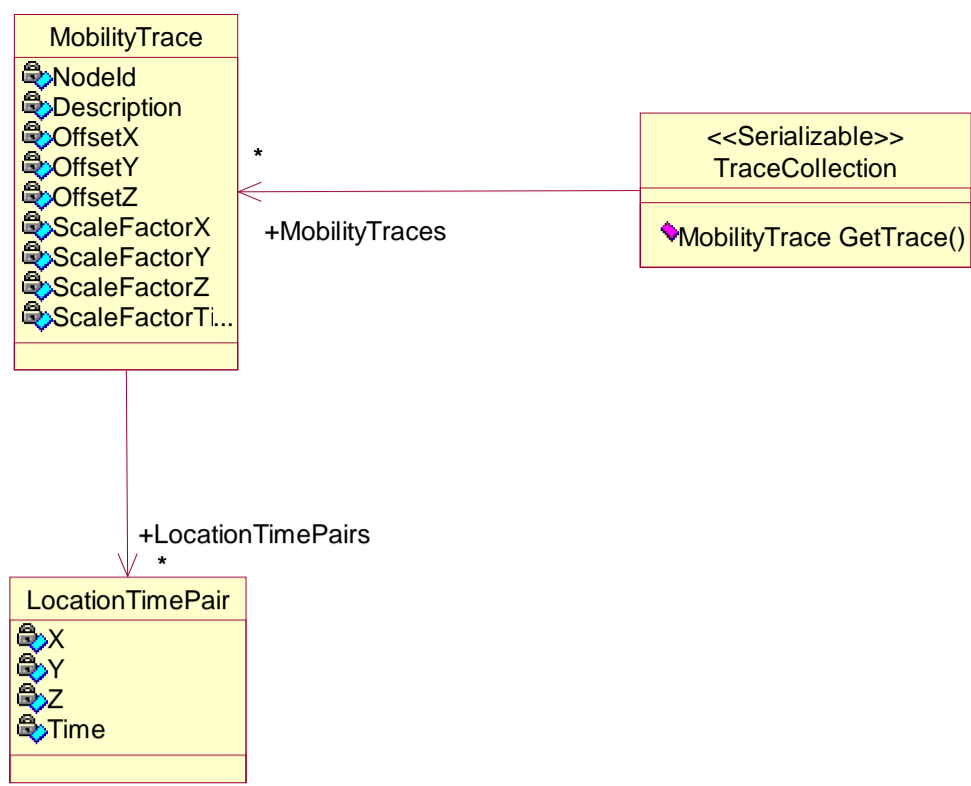


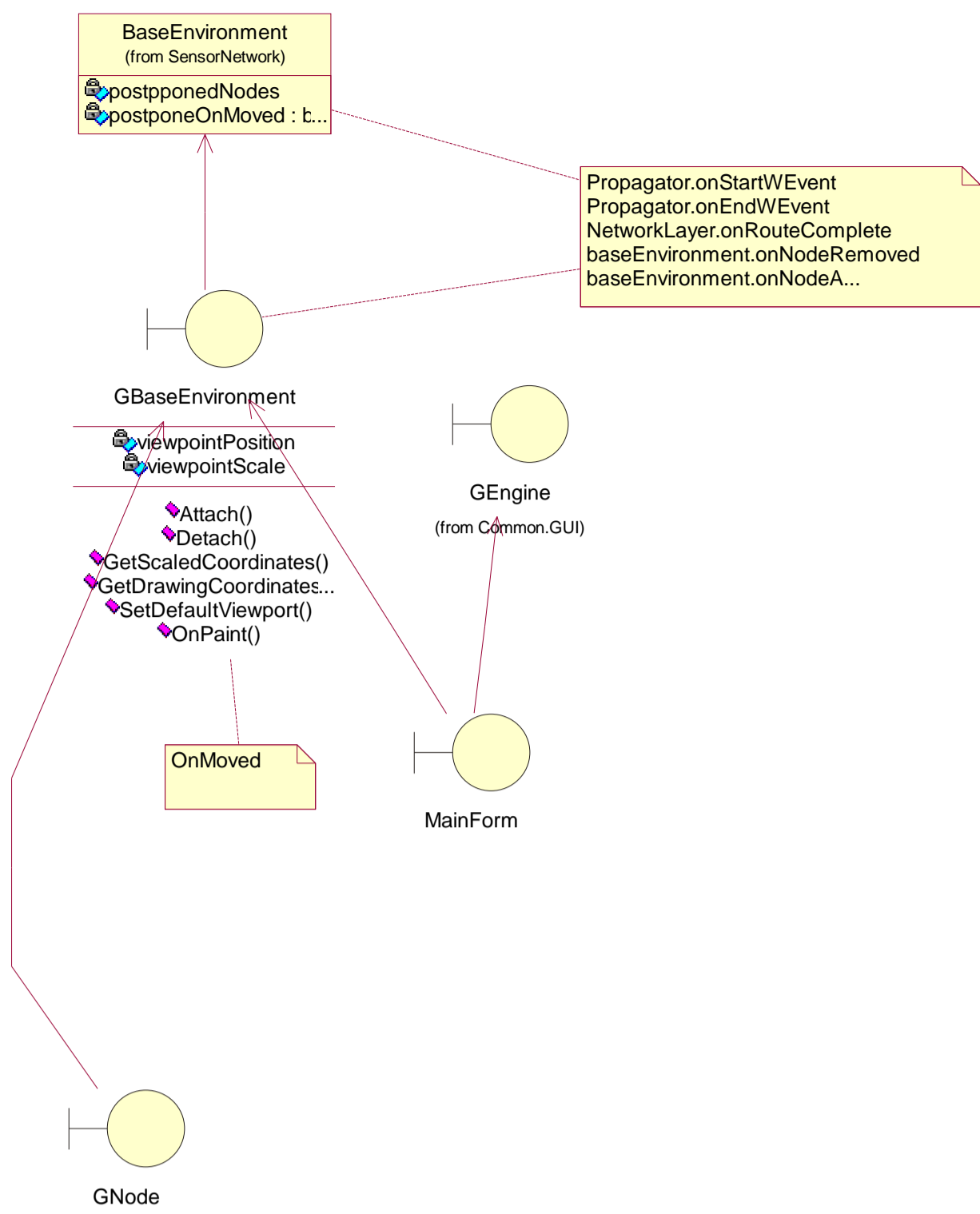


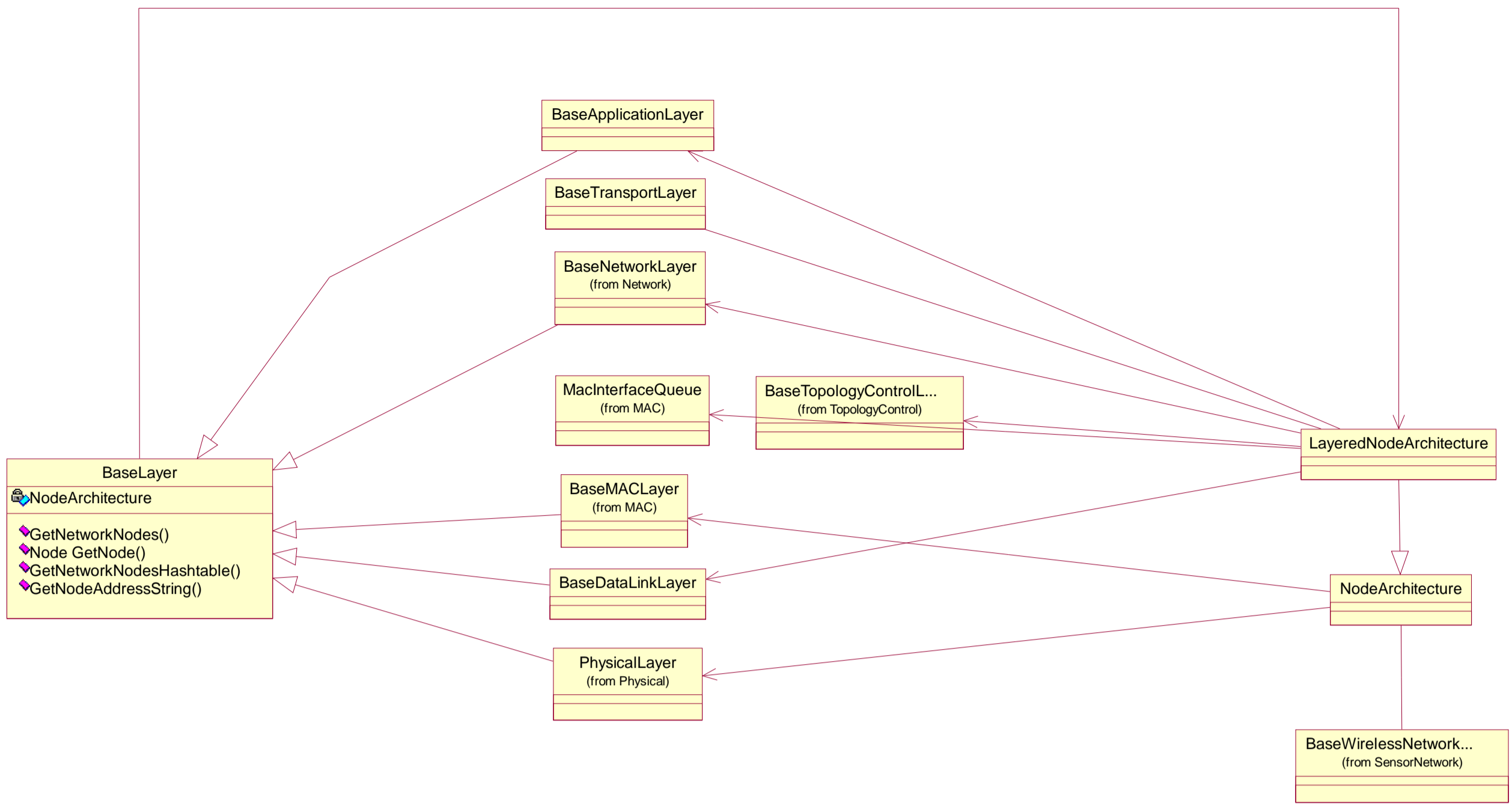


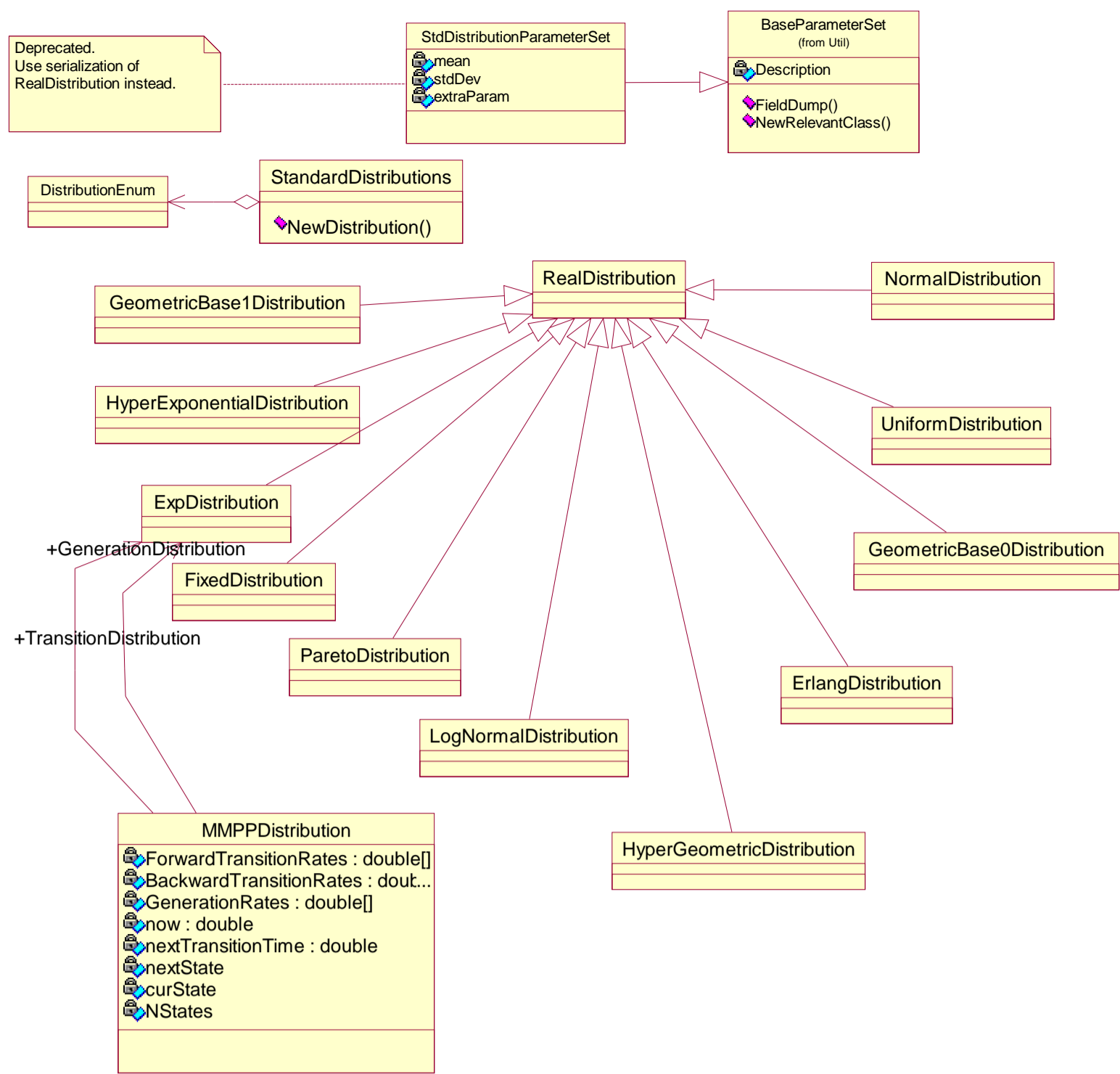


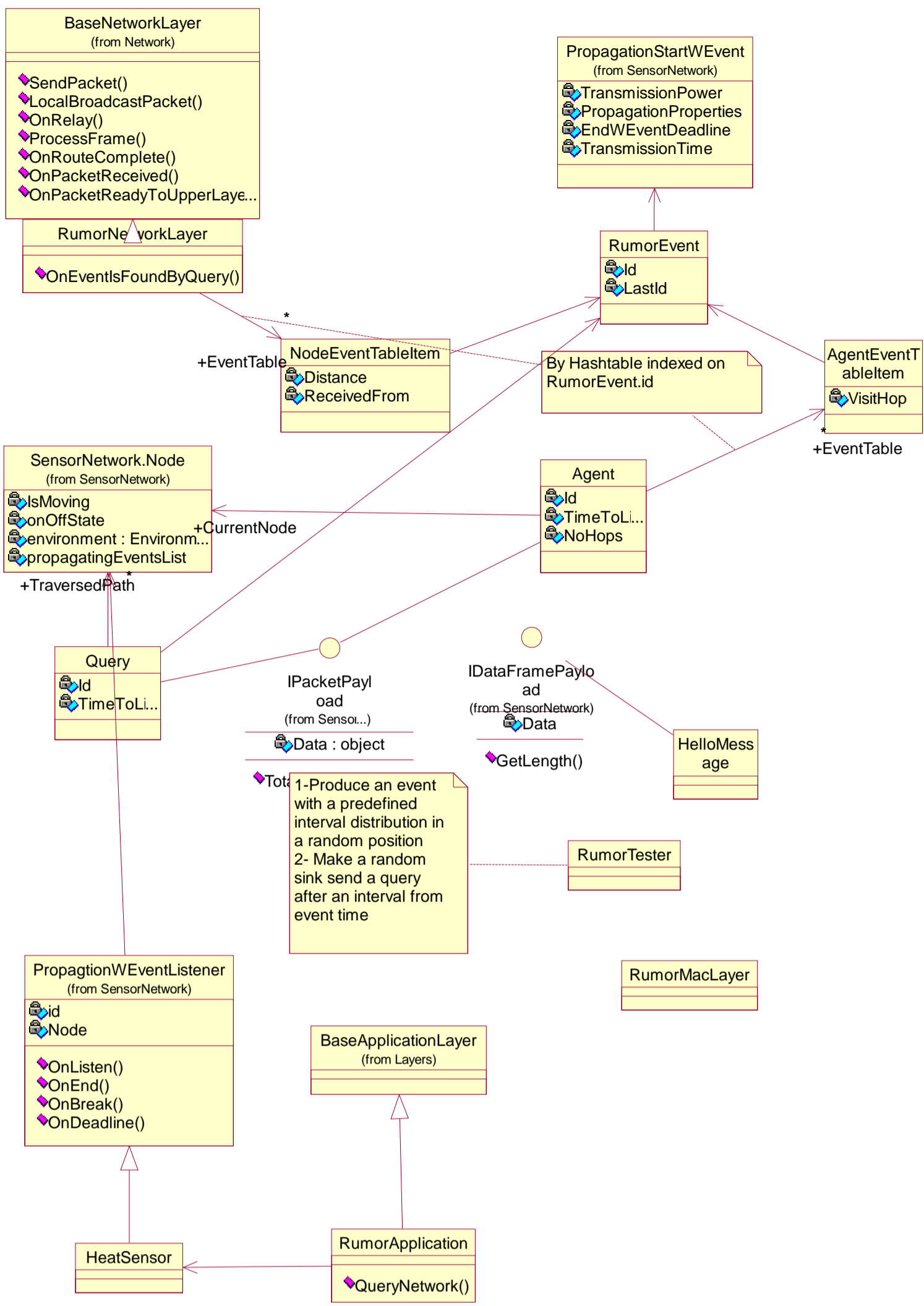


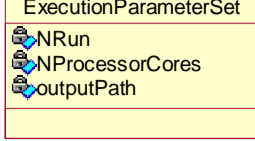
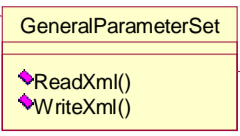
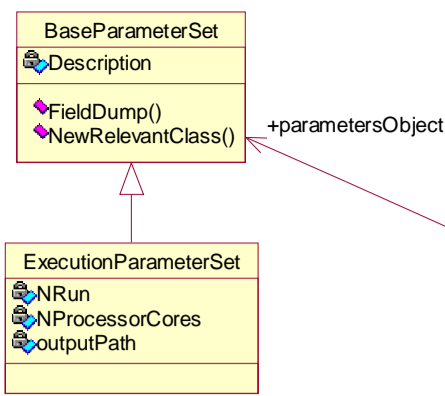












It wraps a BaseParameterSet class. Using a XML attribute named "parameterClassType" the strong type of the wrapped class is identified. When a XML tag of this kind is read, using this attribute the inner XML tag is deserialized into an object. Then, using the NewRelevantClass() method of the object the main class is built.

As an example, to get parameters of a general network layer we define:

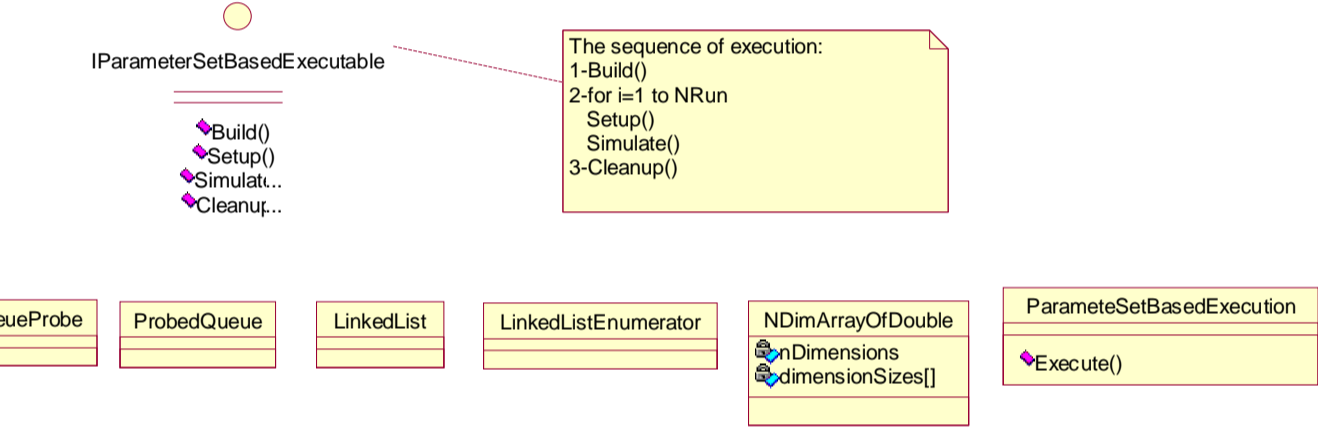
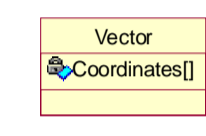
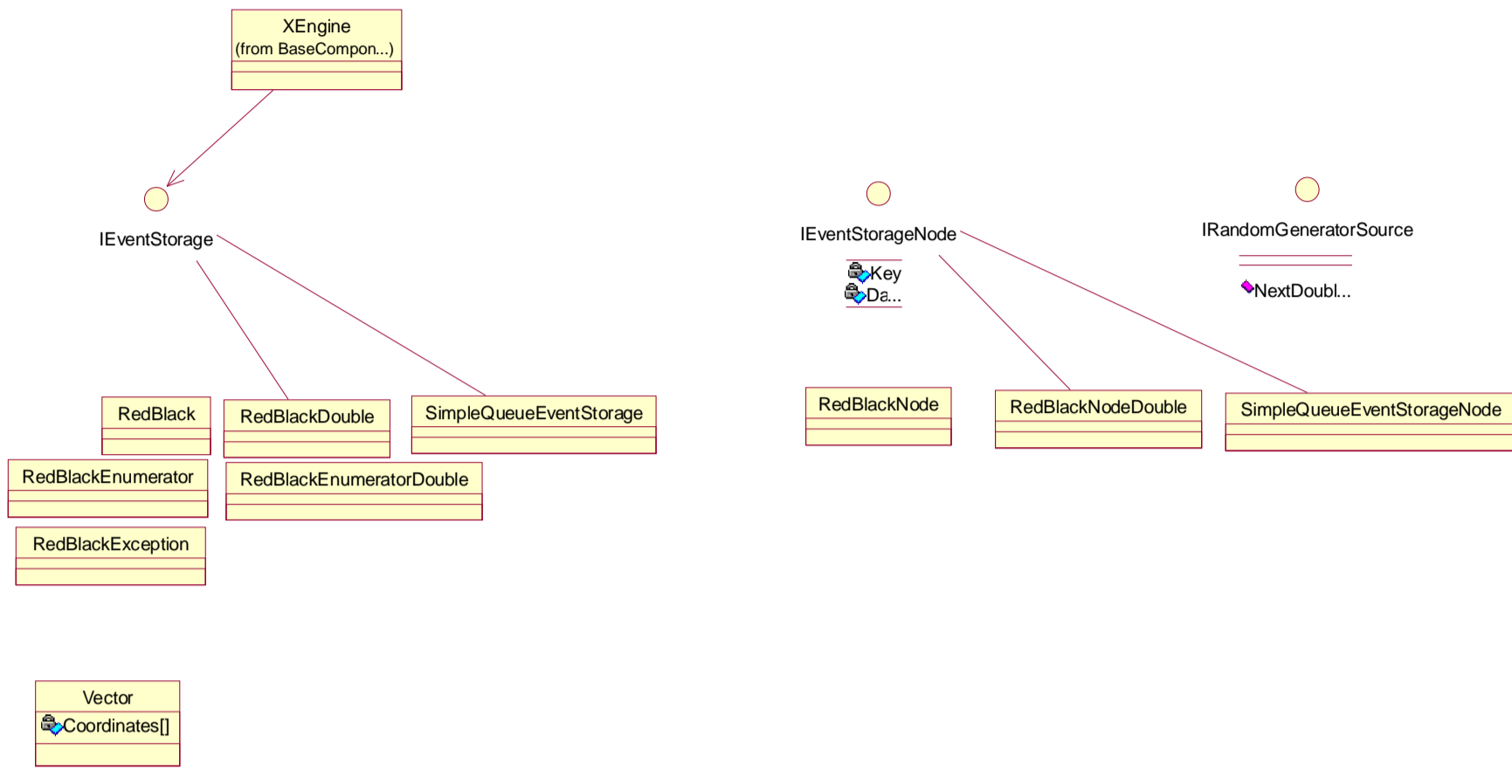
```
public GeneralParameterSet NetworkLayerGParameterSet = null;
```

The XML tag will be:

```
<NetworkLayerGParameterSet parameterClassType="RandomRouting.RandomRoutingNetworkLayerParameterSet, RandomRouting">
  <RandomRoutingNetworkLayerParameterSet>
    <id>Set1</id>
    <description />
    <TTL>20</TTL>
  </RandomRoutingNetworkLayerParameterSet>
</NetworkLayerGParameterSet>
```

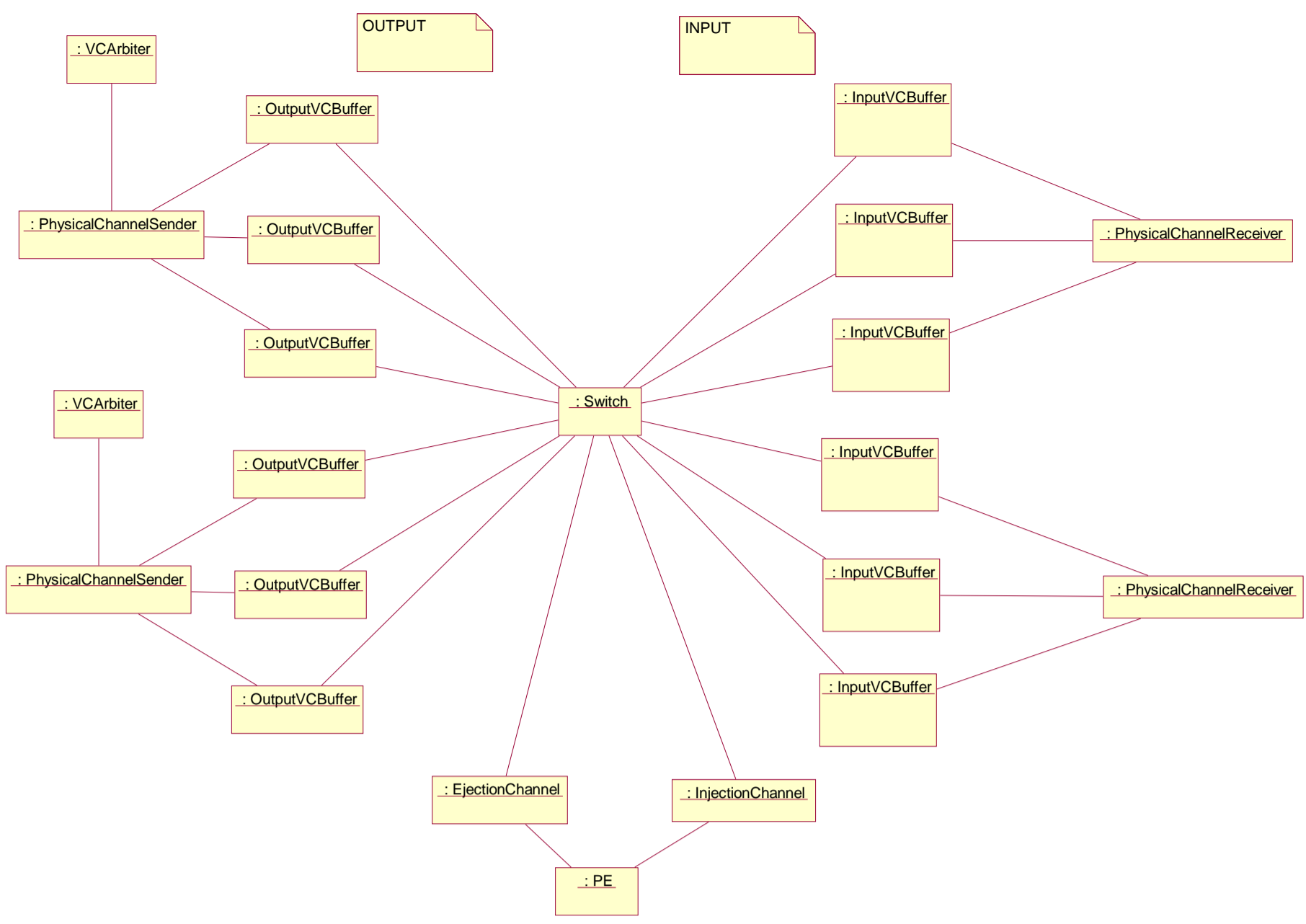
When the tag is read, we can generate the corresponding network layer by

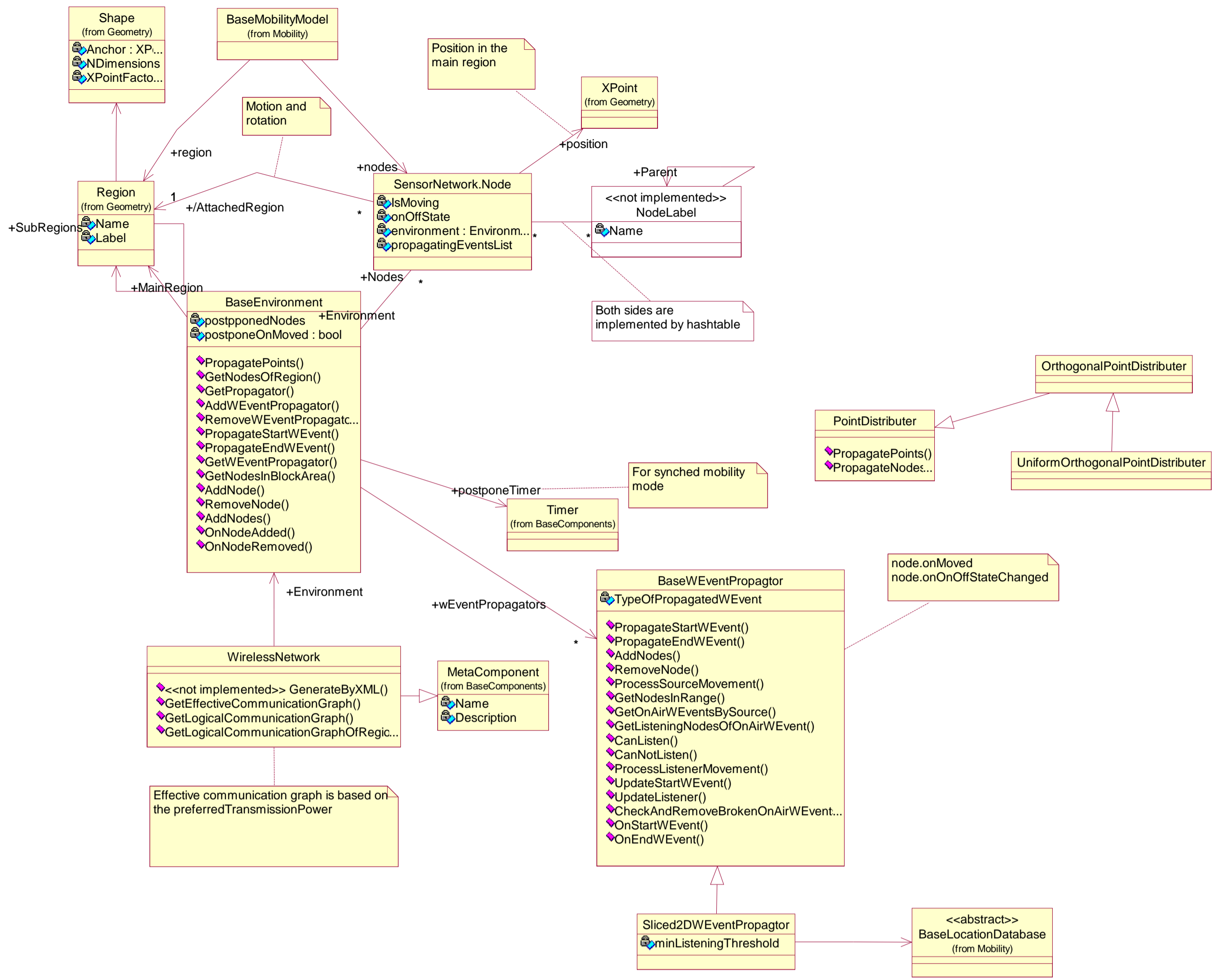
```
netLayer = (BaseNetworkLayer)(parameters.NetworkLayerGParameterSet.parametersObject.NewRelevantClass(nodeld + ".Net"));
```



The sequence of execution:  
 1-Build()  
 2-for i=1 to NRun  
 Setup()  
 Simulate()  
 3-Cleanup()









BaseWirelessNetworkNode.NodeArchitecture.PhysicalLayer.Antenna==  
BaseWirelessNetworkNode.PropagationWEventListeners <=>  
Listener.Node

