```
smacc::ISmaccClientBehavior
- stateMachine
currentState
- currentOrthogonal
+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ getNode()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# getCurrentState()
# executeOnEntry()
# executeOnExit()
# dispose()
- onOrthogonalAllocation()
             Δ
smacc::SmaccClientBehavior
+ onEntry()
+ onExit()
             Δ
                   cl_multirole_sensor
                   ::CbDefaultMultiRoleSensor
                     Behavior < ClientType >
                   + sensor
                   + deferedEventPropagation

    CbDefaultMultiRoleSensor

                   Behavior()
                   + onOrthogonalAllocation()
                   + propagateEvent()
                   + propagateEvent2()
                   + onEntry()
                   + onExit()
                   + onMessageCallback()
                   + getEventLabel()
                         < ClLidarSensor >
   multirole_sensor
 ::CbDefaultMultiRoleSensor
 Behavior < ClLidarSensor >
 + sensor
 + deferedEventPropagation
 + CbDefaultMultiRoleSensor
 Behavior()
 + onOrthogonalAllocation()
 + propagateEvent()
 + propagateEvent2()
 + onEntry()
 + onExit()
 + onMessageCallback()
 + getEventLabel()
              Д
             bot::cl lidar
   sm dance
       ::CbLidarSensor
   + CbLidarSensor()
   + onEntry()
   + onMessageCallback()
```