```
smacc::ISmaccClient
     # components_
     - stateMachine
     + ISmaccClient()
     + ~ISmaccClient()
     + initialize()
     + setStateMachine()
     + getName()
     + postEvent()
     + postEvent()
     + getComponent()
     + configureEventSourceTypes()
     + getType()
     + getStateMachine()
                    smacc::client_bases
                    ::SmaccSubscriberClient
                           < MessageType >
                     + topicName
                     + queueSize
                     + onFirstMessageReceived_
                     + onMessageReceived
                    + postMessageEvent
                     + postInitialMessageEvent
                    # nh
                    - sub

    firstMessage_

                    - initialized
                    + SmaccSubscriberClient()
                     + ~SmaccSubscriberClient()
                    + onMessageReceived()
                     + onFirstMessageReceived()
                     + configureEventSourceTypes()
                     + initialize()
                    - messageCallback()
                    < sensor_msgs::LaserScan >
smacc::client_bases
::SmaccSubscriberClient
 < sensor_msgs::LaserScan >
                                     multirole sensor client
+ topicName
                                     ::ClMultiroleSensor< Message
+ queueSize
                                                 Type >
+ onFirstMessageReceived_
                                     + onMessageTimeout
+ onMessageReceived_
                                     + postTimeoutMessageEvent
+ postMessageEvent
+ postInitialMessageEvent
                                     + timeout
# nh_
                                     timeoutTimer_
- sub
                                     - initialized

    firstMessage

                                     + ClMultiroleSensor()

    initialized

                                     + onMessageTimeout()
+ SmaccSubscriberClient()
                                     + configureEventSourceTypes()
+ ~SmaccSubscriberClient()
                                     + initialize()
+ onMessageReceived()
                                     # resetTimer()
+ onFirstMessageReceived()
                                     timeoutCallback()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
                                       < sensor_msgs::LaserScan >
           multirole_sensor_client
           ::ClMultiroleSensor< sensor
                _msgs::LaserScan >
           + onMessageTimeout
           + postTimeoutMessageEvent
           + timeout

    timeoutTimer

    initialized

           + ClMultiroleSensor()
           + onMessageTimeout()
           + configureEventSourceTypes()
           + initialize()
           # resetTimer()
           timeoutCallback()
               sm dance bot::cl lidar
                  ::ClLaserSensor
```

+ ClLaserSensor()