smacc::SmaccClientBehavior + stateMachine + currentState + currentOrthogonal + SmaccClientBehavior() + ~SmaccClientBehavior() + getName() + onEntry() + onExit() + postEvent() + postEvent() + requiresClient() + requiresComponent() configureEventSourceTypes() multirole sensor client ::CbDefaultMultiRoleSensorBehavior < ClientType > + sensor + deferedEventPropagation + CbDefaultMultiRoleSensor Behavior() + configureEventSourceTypes() + propagateEvent() + propagateEvent2() + onEntry() + onExit() + onMessageCallback() + getEventLabel() < CITemperatureSensor > < CILaserSensor > multirole sensor client multirole sensor client ::CbDefaultMultiRoleSensorBehavior ::CbDefaultMultiRoleSensorBehavior < CITemperatureSensor > < ClLaserSensor > + sensor + sensor + deferedEventPropagation + deferedEventPropagation + CbDefaultMultiRoleSensor + CbDefaultMultiRoleSensor Behavior() Behavior() + configureEventSourceTypes() + configureEventSourceTypes() + propagateEvent() + propagateEvent() + propagateEvent2() + propagateEvent2() + onEntry() + onEntry() + onExit() + onExit() + onMessageCallback() + onMessageCallback() + getEventLabel() + getEventLabel() Δ sm dance bot::cl lidar sm dance bot::cl temperature ::CbLidarSensor _sensor::CbConditionTemperatureSensor + CbLidarSensor() + CbConditionTemperatureSensor() + onEntry() + onMessageCallback() + onMessageCallback()