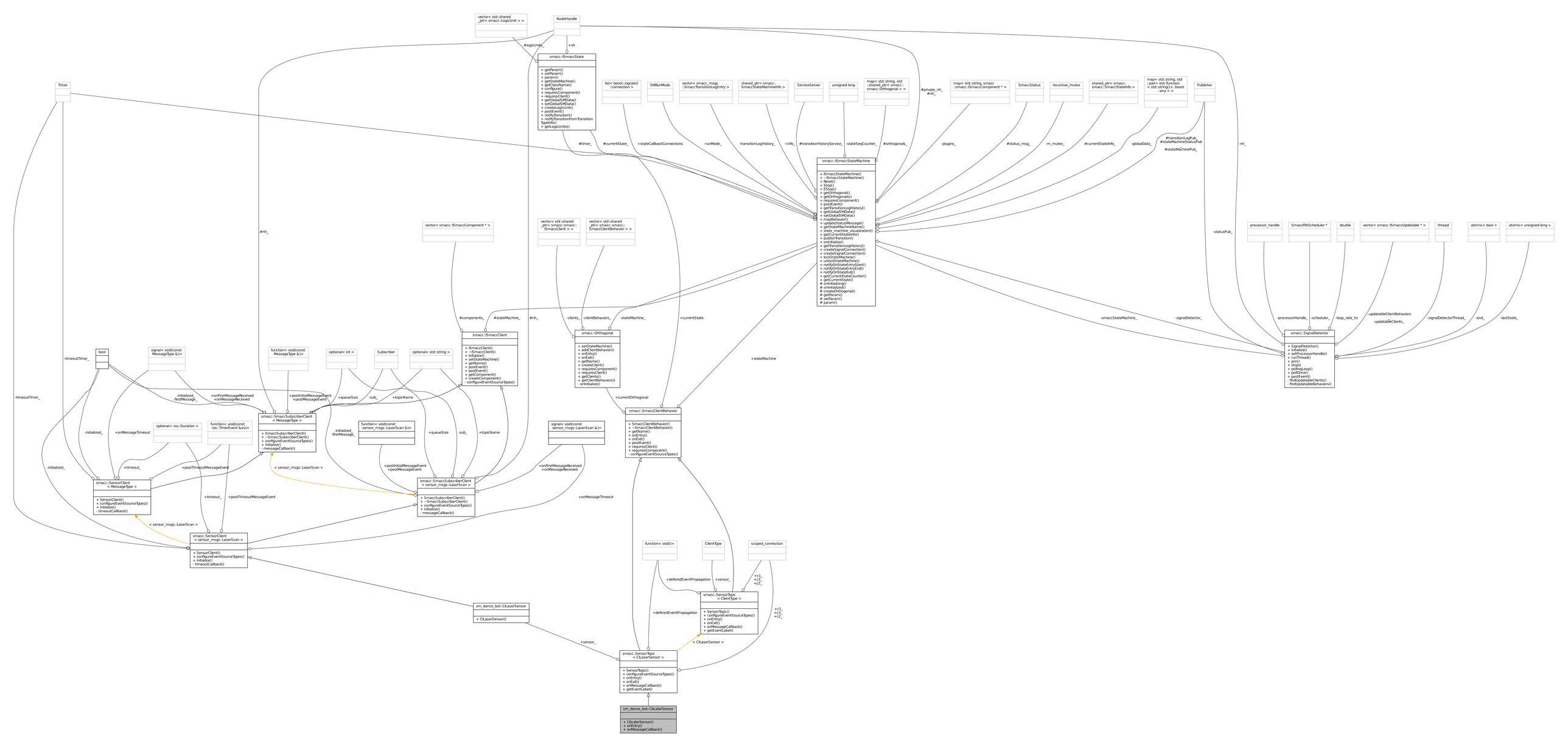
vector< std::shared _ptr< smacc::LogicUnit > >
NodeHandle
#logicUnits_
+nh
smacc::ISmaccState
+ getParam()
+ setParam()
+ params()
+ getStateMachine()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeEntry()
+ getLogicUnits()
Timer
list< boost::signals2 ::connection >
SMRunMode
vector< smacc_msgs ::SmaccTransitionLogEntry >
shared_ptr< smacc:: SmaccStateMachineInfo >
ServiceServer
unsigned long
map< std::string, std ::shared_ptr< smacc:: smacc::IOrthogonal > >
#private_nh_
#nh_
map< std::string, smacc ::smacc::ISmaccComponent * >
SmaccStatus
recursive_mutex
shared_ptr< smacc:: smacc::ISmaccStateInfo >
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
Publisher
#timer_
#currentState_
+stateCallbackConnections
-runMode_
-transitionLogHistory_
+info_
#transitionHistoryService_
-stateSeqCounter_
#orthogonals_
-plugins_
#status_msg_
-m_mutex_
#currentStateInfo_
-globalData_
#transitionLogPub
#stateMachineStatusPub
#stateMachinePub_
-nh_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ Reset()
+ SStop()
+ EStop()
+ getOrthogonal()
+ getOrthogonals()
+ getOrthoComponent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ getStateMachineName()
+ state_machine_visualization()
+ publishStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ createSignalConnection()
+ lockStateMachine()
+ unlockStateMachine()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnStateExitEnd()
+ getCurrentStateCounter()
+ getCurrentState()
# oninitializing()
# oninitialized()
# createOrthogonal()
# getParam()
# setParam()
# params()
vector< smacc::ISmaccComponent * >
vector< std::shared _ptr< smacc::smacc: ISmaccClient > >
vector< std::shared _ptr< smacc::smacc: SmaccClientBehavior > >
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
-statusPub_
#components_
#stateMachine_
#nh_
-clients_
-clientBehaviors_
-stateMachine_
+currentState
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ getEvent()
+ getComponent()
+ createComponent()
- configureEventSourceTypes()
smacc::Orthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ getName()
+ createClient()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
- oninitialize()
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ run()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
bool
-timeoutTimer_
signal< void(const MessageType &)>
function< void(const MessageType &)>
optional< int >
Subscriber
optional< std::string >
+stateMachine
-initialized_
-firstMessage_
+onFirstMessageReceived
+onMessageReceived
+postInitialMessageEvent
+postMessageEvent
+queueSize
-sub_
+topicName
+currentOrthogonal
smacc::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ getClient()
+ requiresComponent()
- configureEventSourceTypes()
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
function< void(const sensor_msgs::LaserScan &)>
signal< void(const sensor_msgs::LaserScan &)>
+onMessageTimeout
+timeout_
+postTimeoutMessageEvent
< sensor_msgs::LaserScan >
smacc::SmaccSubscriberClient < sensor_msgs::LaserScan >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SensorClient < MessageType >
+ SensorClient()
+ configureEventSourceTypes()
+ initialize()
- timeoutCallback()
smacc::SensorClient < sensor_msgs::LaserScan >
+ SensorClient()
+ configureEventSourceTypes()
+ initialize()
- timeoutCallback()
function< void()>
ClientType
scoped_connection
+deferredEventPropagation
+sensor_
+c1_
+c2_
smacc::SensorTopic < ClientType >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::CILaserSensor
+ CILaserSensor()
+sensor_
< CILaserSensor >
smacc::SensorTopic < CILaserSensor >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::CIbLidarSensor
+ CIbLidarSensor()
+ onEntry()
+ onMessageCallback()