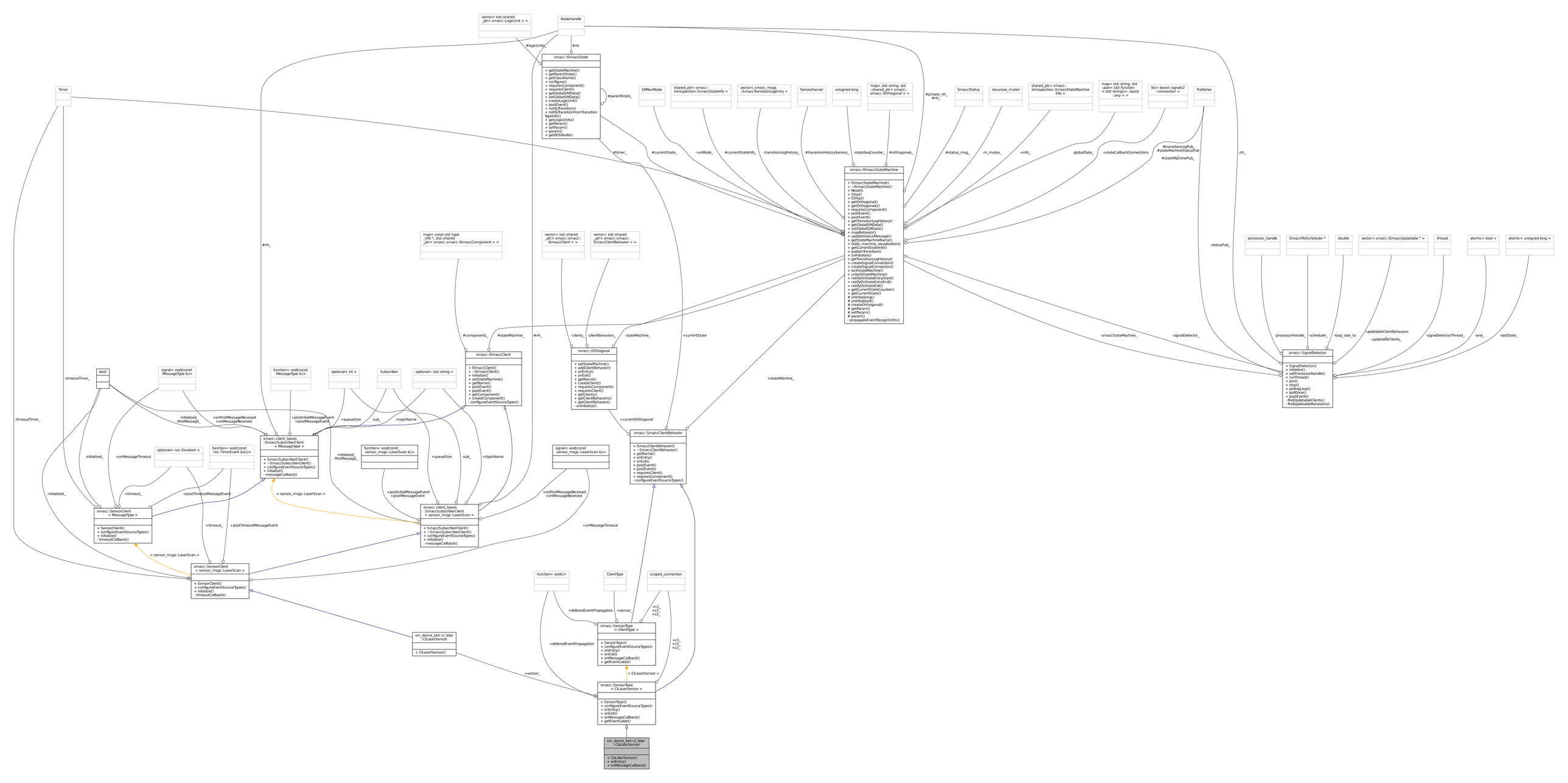
vector< std::shared _ptr< smacc::LogicUnit > >
NodeHandle
#logicUnits_
#nh
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ getOrthogonal()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getLogicUnits()
+ getParam()
+ setParam()
+ param()
+ getROSNode()
#parentState_
Timer
SMRunMode
shared_ptr< smacc:: introspection::SmaccStateInfo >
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
unsigned long
map< std::string, std ::shared_ptr< smacc:: smacc::IOrthogonal > >
#private_nh_ #nh_
SmaccStatus
recursive_mutex
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
list< boost::signals2 ::connection >
Publisher
#timer_
#currentState_
-runMode_
#currentStateInfo_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
#orthogonals_
#status_msg_
-m_mutex_
+info_
-globalData_
+stateCallbackConnections
#transitionLogPub_
#stateMachineStatusPub_
#stateMachinePub_
-nh_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ Reset()
+ Stop()
+ EStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ getParam()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ lockTransition()
+ onInitialized()
+ getTransitionLogHistory()
+ createSignalConnection()
+ createSignalConnection()
+ lockStateMachine()
+ unlockStateMachine()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnStateExit()
+ getCurrentStateCounter()
# onInitialize()
# onInitializing()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- propagateEventToLogicUnits()
-statusPub_
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
map< const std::type _info * std::shared _ptr< smacc::ISmaccComponent > >
vector< std::shared _ptr< smacc::smacc:: ISmaccClient > >
vector< std::shared _ptr< smacc::smacc:: SmaccClientBehavior > >
#components_
#stateMachine_
#nh_
-clients_
-clientBehaviors_
-stateMachine_
+currentState
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findSmaccClients()
- findUpdatableBehaviors()
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ postEvent()
+ getComponent()
+ createComponent()
- configureEventSourceTypes()
smacc::Orthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ postEvent()
+ createClient()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ getClientBehavior()
- onInitialize()
+stateMachine_
+currentOrthogonal
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ getClient()
+ requiresComponent()
- configureEventSourceTypes()
-timeoutTimer_
bool
signal< void(const MessageType &)>
function< void(const MessageType &)>
optional< int >
Subscriber
optional< std::string >
-initialized_ -firstMessage_
+onFirstMessageReceived +onMessageReceived
+postInitialMessageEvent +postMessageEvent
+queueSize
-sub_
+topicName
smacc::client_bases :SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
function< void(const sensor_msgs::LaserScan &)>
signal< void(const sensor_msgs::LaserScan &)>
-initialized_ -firstMessage_
+onMessageTimeout
+timeout_
+postTimeoutMessageEvent
< sensor_msgs::LaserScan >
smacc::client_bases :SmaccSubscriberClient + sensor_msgs::LaserScan >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
+onFirstMessageReceived +onMessageReceived
smacc::SensorClient < MessageType >
+ SensorClient()
+ configureEventSourceTypes()
+ initialize()
- timeoutCallback()
smacc::SensorClient < sensor_msgs::LaserScan >
+ SensorClient()
+ configureEventSourceTypes()
+ initialize()
- timeoutCallback()
function< void()>
ClientType
scoped_connection
+deferedEventPropagation
+sensor_
+c1_ +c3_ +c2_
smacc::SensorTopic < ClientType >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_lidar ::CILaserSensor
+ CILaserSensor()
+sensor_
< CILaserSensor >
smacc::SensorTopic < CILaserSensor >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_lidar ::CbLidarSensor
+ CbLidarSensor()
+ onEntry()
+ onMessageCallback()