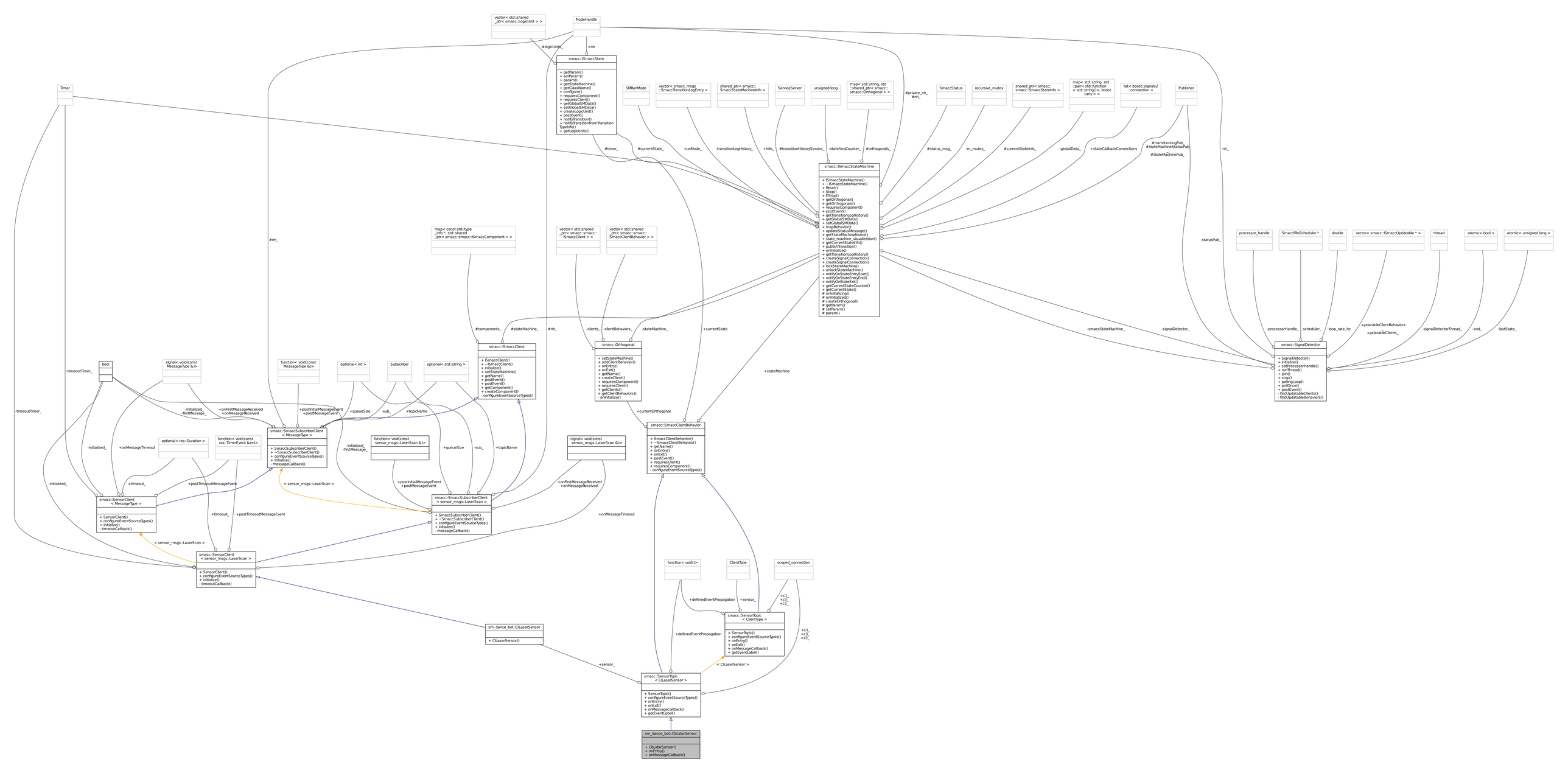
vector< std::shared_ptr< smacc::LogicUnit > >
NodeHandle
#logicUnits_
+nh
smacc::ISmaccState
+ getParam()
+ setParam()
+ param()
+ getStateMachine()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ getTransitionFromTransitionTypeInfo()
+ getLogicUnits()
Timer
SMRunMode
vector< smacc_msgs::SmaccTransitionLogEntry >
shared_ptr< smacc::SmaccStateMachineInfo >
ServiceServer
unsigned long
map< std::string, std::shared_ptr< smacc::smacc::Orthogonal > >
#private_nh_
#nh_
SmaccStatus
recursive_mutex
shared_ptr< smacc::smacc::SmaccStateMachineInfo >
map< std::string, std::pair< std::function< std::string()>, boost::any > >
list< boost::signals2::connection >
Publisher
#timer_
#currentState_
-runMode_
-transitionLogHistory_
+info_
#transitionHistoryService_
-stateSeqCounter_
#orthogonals_
#status_msg_
-m_mutex_
#currentStateInfo_
-globalData_
+stateCallbackConnections
#TransitionLogPub_
#stateMachineStatusPub_
#stateMachinePub_
-nh_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ Reset()
+ EStop()
+ EStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ getStateMachineName()
+ state_machine_visualization()
+ publishStateMachine()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ createSignalConnection()
+ lockStateMachine()
+ unlockStateMachine()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnStateExit()
+ getCurrentStateCounter()
+ getCurrentState()
# getInitialStates()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
-statusPub_
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
map< const std::type_info *, std::shared_ptr< smacc::smacc::ISmaccComponent > >
vector< std::shared_ptr< smacc::smacc::ISmaccClient > >
vector< std::shared_ptr< smacc::smacc::ISmaccClientBehavior > >
#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()
- findUpdatableClients()
- findUpdatableBehaviors()
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ requiresComponent()
+ createComponent()
- configureEventSourceTypes()
smacc::Orthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ getName()
+ createClient()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
- onInitialize()
+currentOrthogonal
+stateMachine
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ requiresClient()
+ 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::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
+onMessageTimeout
+timeout_
+postTimeoutMessageEvent
function< void(const sensor_msgs::LaserScan &)>
< sensor_msgs::LaserScan >
signal< void(const 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()
sm_dance_bot::CILaserSensor
+ CILaserSensor()
function< void()>
ClientType
scoped_connection
+deferedEventPropagation
+sensor_
+c1_
+c3_
+c2_
smacc::SensorTopic < ClientType >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
< CILaserSensor >
smacc::SensorTopic < CILaserSensor >
+ SensorTopic()
+ configureEventSourceTypes()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::CbLidarSensor
+ CbLidarSensor()
+ onEntry()
+ onMessageCallback()