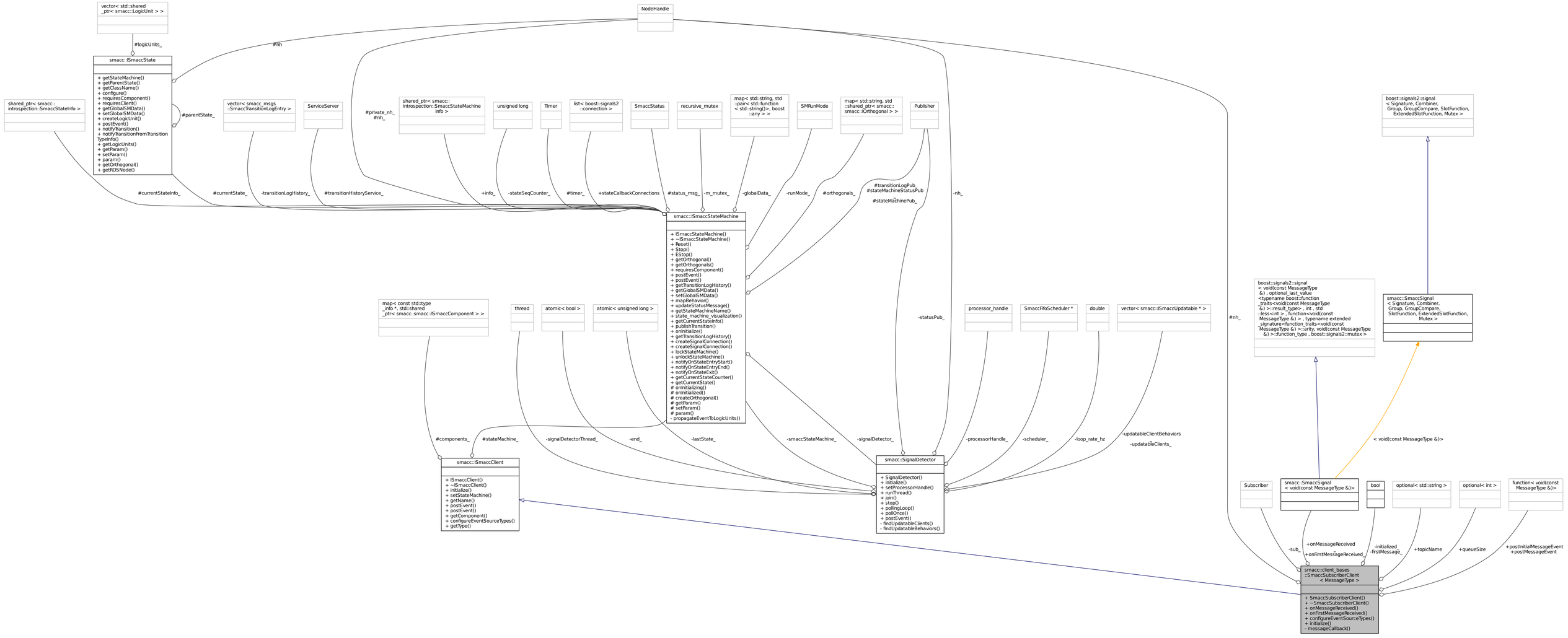
vector< std::shared _ptr< smacc::LogicUnit > >
#logicUnits_
NodeHandle
#nh
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getLogicUnits()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getROSNode()
#parentState_
shared_ptr< smacc:: introspection::SmaccStateInfo >
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
#private_nh_
#nh_
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
unsigned long
Timer
list< boost::signals2 ::connection >
SmaccStatus
recursive_mutex
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
SMRunMode
map< std::string, std ::shared_ptr< smacc:: smacc::IOrthogonal > >
Publisher
#currentStateInfo_
#currentState_
-transitionLogHistory_
#transitionHistoryService_
+info_
-stateSeqCounter_
#timer_
+stateCallbackConnections
#status_msg_
-m_mutex_
-globalData_
-runMode_
#orthogonals_
#transitionLogPub
#stateMachineStatusPub
#stateMachinePub_
-nh_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ Reset()
+ Stop()
+ EStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ oninitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ createSignalConnection()
+ lockStateMachine()
+ unlockStateMachine()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnStateExit()
+ getCurrentStateCounter()
+ getCurrentState()
# oninitializing()
# oninitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- propagateEventToLogicUnits()
map< const std::type _info *, std::shared _ptr< smacc::smacc::ISmaccComponent > >
thread
atomic< bool >
atomic< unsigned long >
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
-statusPub_
#components_
#stateMachine_
-signalDetectorThread_
-end_
-lastState_
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
boost::signals2::signal < void(const MessageType &) , optional, last, value <typename boost::function traits<void(const MessageType &) >::result_type> , int , std ::less<int > , function<void(const MessageType &) > , typename extended _signature<function_traits<void(const MessageType &) >::arity, void(const MessageType &) >::function_type , boost::signals2::mutex >
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
#nh_
< void(const MessageType &)>
Subscriber
smacc::SmaccSignal < void(const MessageType &)>
bool
optional< std::string >
optional< int >
function< void(const MessageType &)>
-sub_
+onMessageReceived
+onFirstMessageReceived
-initialized_
-firstMessage_
+topicName
+queueSize
+postInitialMessageEvent
+postMessageEvent
smacc::client_bases ::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()