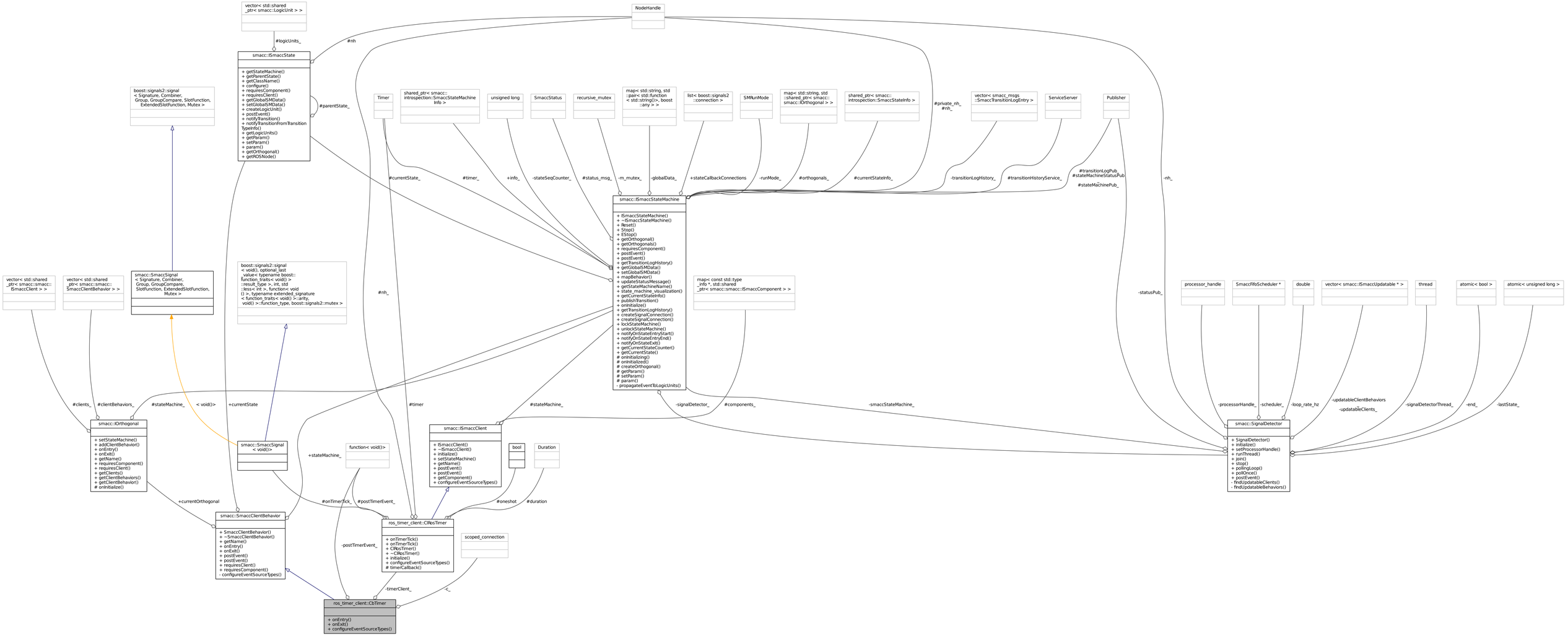
NodeHandle
vector< std::shared _ptr< smacc::LogicUnit > >
#logicUnits_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getLogicUnits()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getROSNode()
#nh
#parentState_
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
Timer
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
unsigned long
SmaccStatus
recursive_mutex
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
list< boost::signals2 ::connection >
SMRunMode
map< std::string, std ::shared_ptr< smacc:: smacc::IOrthogonal > >
shared_ptr< smacc:: introspection::SmaccStateInfo >
#private_nh_
#nh_
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
Publisher
#currentState_
#timer_
+info_
-stateSeqCounter_
#status_msg_
-m_mutex_
-globalData_
+stateCallbackConnections
-runMode_
#orthogonals_
#currentStateInfo_
-transitionLogHistory_
#transitionHistoryService_
#transitionLogPub_
#smaccStateMachineStatusPub_
#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()
vector< std::shared _ptr< smacc::smacc:: ISmaccClient > >
vector< std::shared _ptr< smacc::smacc:: SmaccClientBehavior > >
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
boost::signals2::signal < void(), optional_last _value< typename boost:: function_traits< void() > ::result_type >, int, std ::less< int >, function< void () >, typename extended_signature < function_traits< void() >::arity, void() >::function_type, boost::signals2::mutex >
#nh_
map< const std::type _info *, std::shared _ptr< smacc::smacc::ISmaccComponent > >
-statusPub_
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
#clients_
#clientBehaviors_
#stateMachine_
< void()>
+currentState
#timer
#stateMachine_
-signalDetector_
#components_
-smaccStateMachine_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
smacc::IOrthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ getClientBehavior()
# onInitialize()
smacc::SmaccSignal < void()>
function< void()>
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
bool
Duration
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
+currentOrthogonal
+stateMachine_
#onTimerTick_
#postTimerEvent_
#oneshot
#duration
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
- configureEventSourceTypes()
ros_timer_client::CRosTimer
+ onTimerTick()
+ onTimerTick()
+ CRosTimer()
+ ~CRosTimer()
+ initialize()
+ configureEventSourceTypes()
# timerCallback()
scoped_connection
-postTimerEvent_
-timerClient_
-c_
ros_timer_client::CbTimer
+ onEntry()
+ onExit()
+ configureEventSourceTypes()