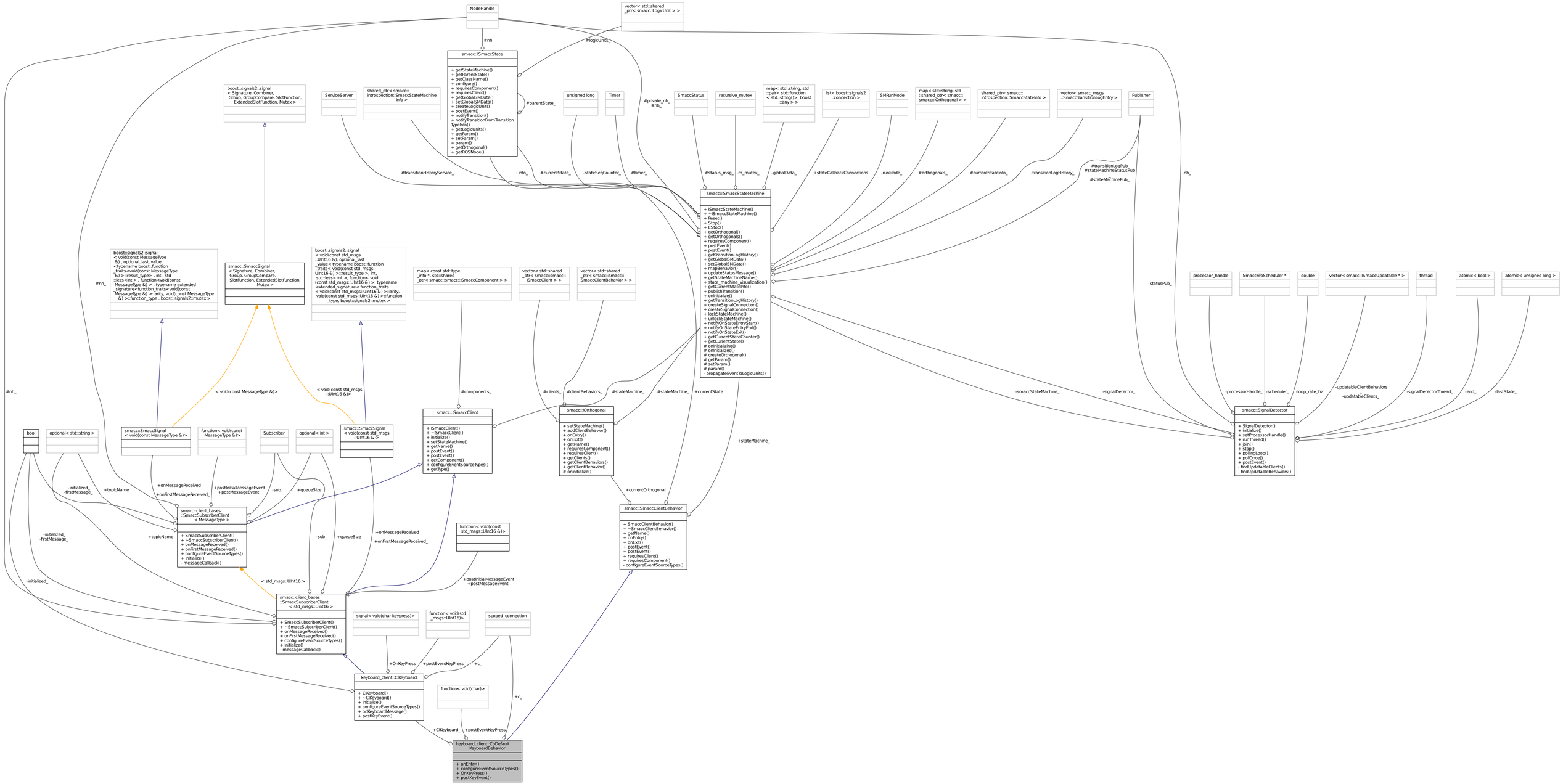
NodeHandle
vector< std::shared _ptr< smacc::LogicUnit > >
#nh
#logicUnits_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getLogicUnits()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getROSNode()
#parentState_
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
ServiceServer
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
unsigned long
Timer
#private_nh_
#nh_
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 >
vector< smacc_msgs ::SmaccTransitionLogEntry >
Publisher
#transitionHistoryService_
+info_
#currentState_
-stateSeqCounter_
#timer_
#status_msg_
-m_mutex_
-globalData_
+stateCallbackConnections
-runMode_
#orthogonals_
#currentStateInfo_
-transitionLogHistory_
#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()
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 >
boost::signals2::signal < void(const std_msgs ::UInt16 &) , optional_last _value< typename boost::function _traits< void(const std_msgs:: UInt16 &) >::result_type > , int, std::less< int >, function< void (const std_msgs::UInt16 &) >, typename extended_signature< function_traits < void(const std_msgs::UInt16 &) >::arity, void(const std_msgs::UInt16 &) >::function _type, boost::signals2::mutex >
map< const std::type _info *, std::shared _ptr< smacc::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_
#nh_
< void(const MessageType &)>
< void(const std_msgs ::UInt16 &)>
#components_
#clients_
#clientBehaviors_
#stateMachine_
#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()
bool
optional< std::string >
smacc::SmaccSignal < void(const MessageType &)>
function< void(const MessageType &)>
Subscriber
optional< int >
smacc::SmaccSignal < void(const std_msgs ::UInt16 &)>
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
smacc::IOrthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ getClientBehavior()
# onInitialize()
+stateMachine_
-initialized_
-firstMessage_
+topicName
+onMessageReceived
+onFirstMessageReceived_
+postInitialMessageEvent
+postMessageEvent
-sub_
+queueSize
+currentOrthogonal
smacc::client_bases ::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
- configureEventSourceTypes()
function< void(const std_msgs::UInt16 &)>
-initialized_
-firstMessage_
+topicName
-sub_
+queueSize
+onMessageReceived
+onFirstMessageReceived_
+postInitialMessageEvent
+postMessageEvent
< std_msgs::UInt16 >
smacc::client_bases ::SmaccSubscriberClient < std_msgs::UInt16 >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
signal< void(char keypress)>
function< void(std _msgs::UInt16)>
scoped_connection
+OnKeyPress
+postEventKeyPress
+c_
keyboard_client::CIKeyboard
+ CIKeyboard()
+ ~CIKeyboard()
+ initialize()
+ configureEventSourceTypes()
+ onKeyboardMessage()
+ postKeyEvent()
function< void(char)>
+c_
+CIKeyboard_
+postEventKeyPress
keyboard_client::CbDefault KeyboardBehavior
+ onEntry()
+ configureEventSourceTypes()
+ OnKeyPress()
+ postKeyEvent()