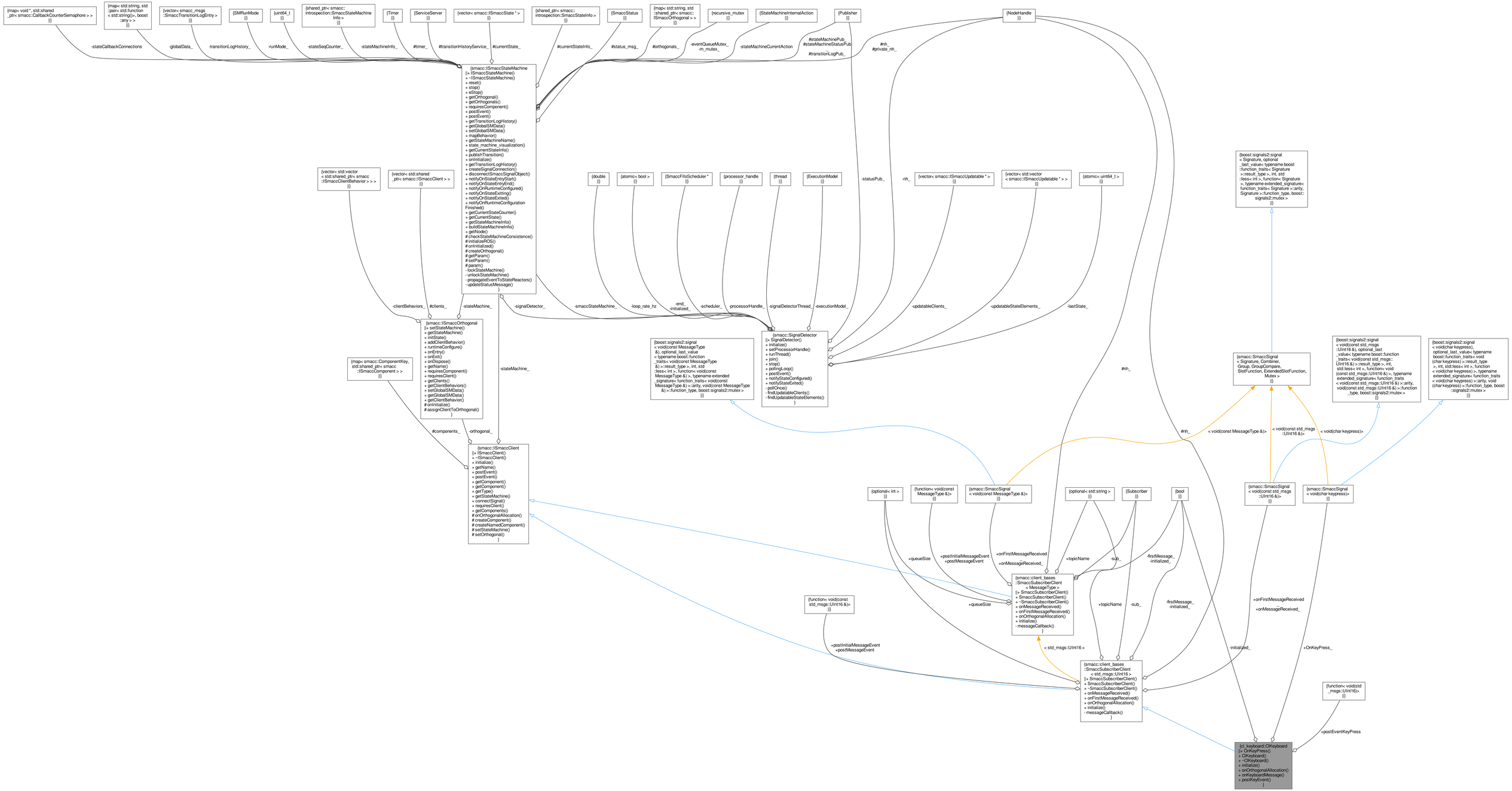
{map< void *, std::shared_ptr< smacc::CallbackCounterSemaphore > > }
{map< std::string, std::pair< std::function < std::string()>, boost ::any > > }
{vector< smacc_msgs ::SmaccTransitionLogEntry > }
{SMRunMode }
{uint64_t }
{shared_ptr< smacc:: introspection::SmaccStateMachine Info > }
{Timer }
{ServiceServer }
{vector< smacc::ISmaccState * > }
{shared_ptr< smacc:: introspection::SmaccStateInfo > }
{SmaccStatus }
{map< std::string, std ::shared_ptr< smacc:: ISmaccOrthogonal > > }
{recursive_mutex }
{StateMachineInternalAction }
{Publisher }
{NodeHandle }
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
-stateMachineInfo_
#timer_
#transitionHistoryService_
#currentState_
#currentStateInfo_
#status_msg_
#orthogonals_
-eventQueueMutex_ -m_mutex_
-stateMachineCurrentAction
#stateMachinePub_ #stateMachineStatusPub_
#transitionLogPub_
#nh_ #private_nh_
{smacc::ISmaccStateMachine
|| + ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ disconnectSmaccSignalObject()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExitting()
+ notifyOnStateExited()
+ notifyOnRuntimeConfiguration Finished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
-statusPub_
-nh_
{vector< smacc::ISmaccUpdatable * > }
{vector< std::vector < smacc::ISmaccUpdatable * > > }
{atomic< uint64_t > }
-updatableClients_
-updatableStateElements_
-lastState_
{vector< std::vector < std::shared_ptr< smacc ::ISmaccClientBehavior > > > }
{vector< std::shared _ptr< smacc::ISmaccClient > > }
{double }
{atomic< bool > }
{SmaccFifoScheduler * }
{processor_handle }
{thread }
{ExecutionModel }
-clientBehaviors_
#clients_
-stateMachine_
-signalDetector_
-smaccStateMachine_
-loop_rate_hz
-end_ -initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
{smacc::ISmaccOrthogonal
|| + setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
}
{map< smacc::ComponentKey, std::shared_ptr< smacc ::ISmaccComponent > > }
-stateMachine_
#components_
-orthogonal_
{smacc::ISmaccClient
|| + ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
}
{boost::signals2::signal
< void(const MessageType &), optional_last_value < typename boost::function < void(const MessageType _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::SignalDetector
|| + SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClients()
- findUpdatableStateElements()
}
{boost::signals2::signal
< Signature, optional _last_value< typename boost ::function_traits< Signature >::result_type >, int, std ::less< int >, function< Signature >, typename extended_signature< function_traits< Signature >::arity, Signature >::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 > }
{boost::signals2::signal
< void(char keypress), optional_last_value< typename boost::function_traits< void (char keypress) >::result_type >, int, std::less< int >, function < void(char keypress) >, typename extended_signature< function_traits < void(char keypress) >::arity, void (char keypress) >::function_type, boost ::signals2::mutex > }
#nh_
< void(const MessageType &)>
< void(const std_msgs ::UInt16 &)>
< void(char keypress)>
{smacc::SmaccSignal
< void(const std_msgs ::UInt16 &)>
{smacc::SmaccSignal
< void(char keypress)>
{optional< int > }
{function< void(const MessageType &)> }
{smacc::SmaccSignal
< void(const MessageType &)>
{optional< std::string > }
{Subscriber }
{bool }
+queueSize
+postInitialMessageEvent +postMessageEvent
+onFirstMessageReceived _
+onMessageReceived_
+topicName
-sub_
-firstMessage_ -initialized_
{smacc::client_bases ::SmaccSubscriberClient < MessageType > || + SmaccSubscriberClient() + SmaccSubscriberClient() + ~SmaccSubscriberClient() + onMessageReceived() + onFirstMessageReceived() + onOrthogonalAllocation() + initialize() - messageCallback() }
{function< void(const std_msgs::UInt16 &)> }
< std_msgs::UInt16 >
{smacc::client_bases ::SmaccSubscriberClient < std_msgs::UInt16 > || + SmaccSubscriberClient() + SmaccSubscriberClient() + ~SmaccSubscriberClient() + onMessageReceived() + onFirstMessageReceived() + onOrthogonalAllocation() + initialize() - messageCallback() }
-initialized_
+onFirstMessageReceived
+onMessageReceived_
+OnKeyPress_
{function< void(std _msgs::UInt16)> }
+postEventKeyPress
{cl_keyboard::ClKeyboard || + OnKeyPress() + ClKeyboard() + ~ClKeyboard() + initialize() + onOrthogonalAllocation() + onKeyboardMessage() + postKeyEvent() }