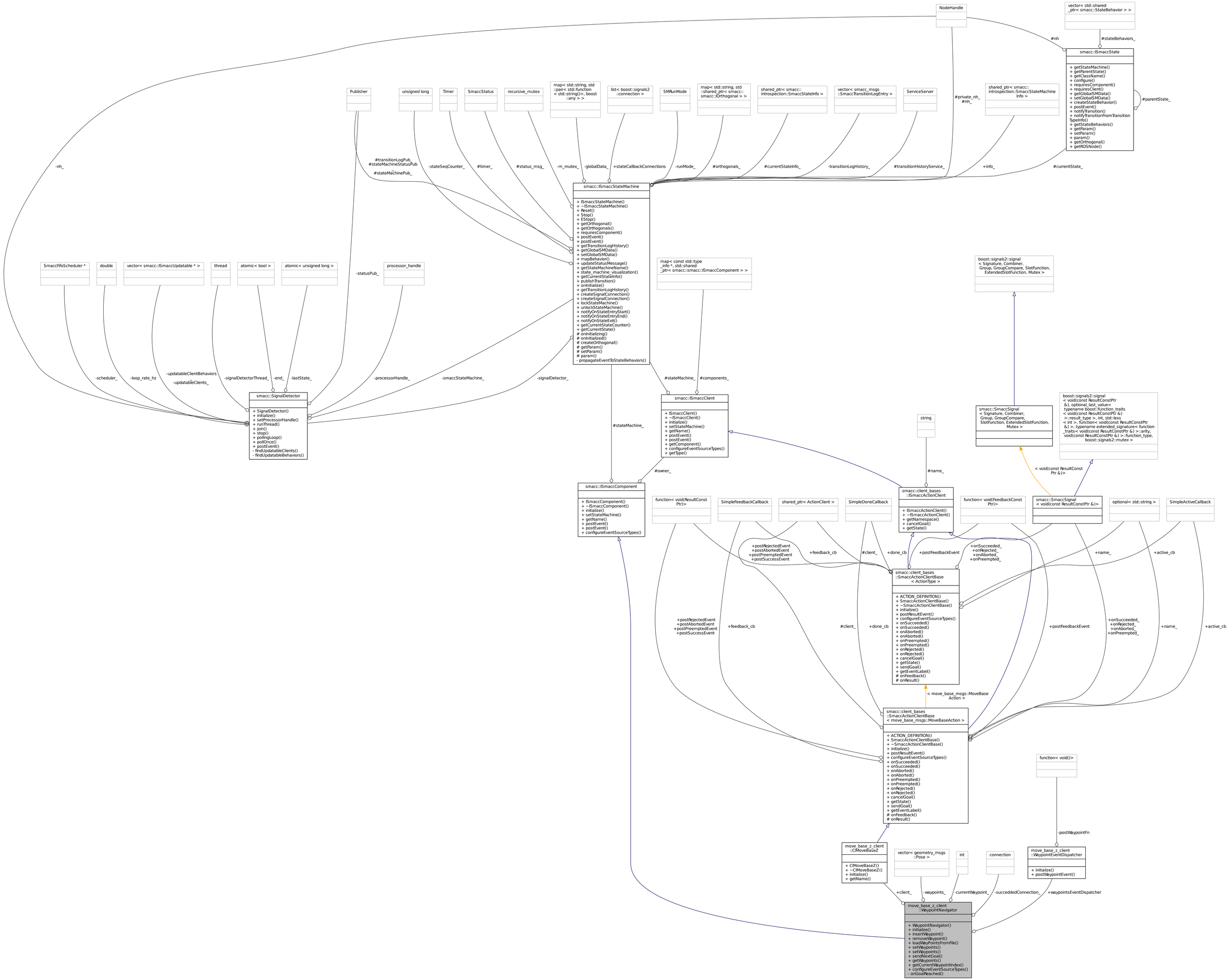
NodeHandle
vector< std::shared _ptr< smacc::StateBehavior > >
#nh
#stateBehaviors_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateBehavior()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateBehaviors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getROSNode()
#parentState_
#private_nh_
#nh_
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
ServiceServer
vector< smacc_msgs ::SmaccTransitionLogEntry >
shared_ptr< smacc:: introspection::SmaccStateInfo >
map< std::string, std ::shared_ptr< smacc:: smacc::IOrthogonal > >
SMRunMode
list< boost::signals2 ::connection >
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
recursive_mutex
SmaccStatus
Timer
unsigned long
Publisher
-nh_
#transitionLogPub_
#stateMachineStatusPub_
#stateMachinePub_
-stateSeqCounter_
#timer_
#status_msg_
-m_mutex_
-globalData_
+stateCallbackConnections
-runMode_
#orthogonals_
#currentStateInfo_
-transitionLogHistory_
#transitionHistoryService_
+info_
#currentState_
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()
- propagateEventToStateBehaviors()
processor_handle
-statusPub_
atomic< unsigned long >
atomic< bool >
thread
vector< smacc::ISmaccUpdatable * >
double
SmaccFifoScheduler *
map< const std::type _info *, std::shared _ptr< smacc::smacc::ISmaccComponent > >
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
-processorHandle_
-smaccStateMachine_
-signalDetector_
#stateMachine_
#components_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
#stateMachine_
#owner_
string
#name_
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
boost::signals2::signal < void(const ResultConstPtr &), optional_last_value< typename boost::function_traits < void(const ResultConstPtr &) >::result_type >, int, std::less < int >, function< void(const ResultConstPtr &) >, typename extended_signature< function _traits< void(const ResultConstPtr &) >::arity, void(const ResultConstPtr &) >::function_type, boost::signals2::mutex >
< void(const ResultConst Ptr &)>
smacc::ISmaccComponent
+ ISmaccComponent()
+ ~ISmaccComponent()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ configureEventSourceTypes()
smacc::client_bases ::ISmaccActionClient
+ ISmaccActionClient()
+ ~ISmaccActionClient()
+ getNamespace()
+ cancelGoal()
+ getState()
function< void(ResultConst Ptr)>
SimpleFeedbackCallback
shared_ptr< ActionClient >
SimpleDoneCallback
function< void(FeedbackConst Ptr)>
smacc::SmaccSignal < void(const ResultConstPtr &)>
optional< std::string >
SimpleActiveCallback
+postRejectedEvent
+postAbortedEvent
+postPreemptedEvent
+postSuccessEvent
+feedback_cb
#client_
+done_cb
+postFeedbackEvent
+onSucceeded_
+onRejected_
+onAborted_
+onPreempted_
+name_
+active_cb
smacc::client_bases ::SmaccActionClientBase < ActionType >
+ ACTION_DEFINITION()
+ SmaccActionClientBase()
+ ~SmaccActionClientBase()
+ initialize()
+ postResultEvent()
+ configureEventSourceTypes()
+ onSucceeded()
+ onSucceeded()
+ onAborted()
+ onAborted()
+ onPreempted()
+ onPreempted()
+ onRejected()
+ onRejected()
+ cancelGoal()
+ getState()
+ sendGoal()
+ getEventLabel()
# onFeedback()
# onResult()
< move_base_msgs::MoveBase Action >
smacc::client_bases ::SmaccActionClientBase < move_base_msgs::MoveBaseAction >
+ ACTION_DEFINITION()
+ SmaccActionClientBase()
+ ~SmaccActionClientBase()
+ initialize()
+ postResultEvent()
+ configureEventSourceTypes()
+ onSucceeded()
+ onSucceeded()
+ onAborted()
+ onAborted()
+ onPreempted()
+ onPreempted()
+ onRejected()
+ onRejected()
+ cancelGoal()
+ getState()
+ sendGoal()
+ getEventLabel()
# onFeedback()
# onResult()
function< void()>
-postWaypointFn
move_base_z_client ::ClMoveBaseZ
+ ClMoveBaseZ()
+ ~ClMoveBaseZ()
+ initialize()
+ getName()
vector< geometry_msgs ::Pose >
int
connection
move_base_z_client ::WaypointEventDispatcher
+ initialize()
+ postWaypointEvent()
+client_
-waypoints_
-currentWaypoint_
-succeddedConnection_
+waypointsEventDispatcher
move_base_z_client ::WaypointNavigator
+ WaypointNavigator()
+ initialize()
+ insertWaypoint()
+ removeWaypoint()
+ loadWayPointsFromFile()
+ setWaypoints()
+ setWaypoints()
+ sendNextGoal()
+ getWaypoints()
+ getCurrentWaypointIndex()
+ configureEventSourceTypes()
- onGoalReached()