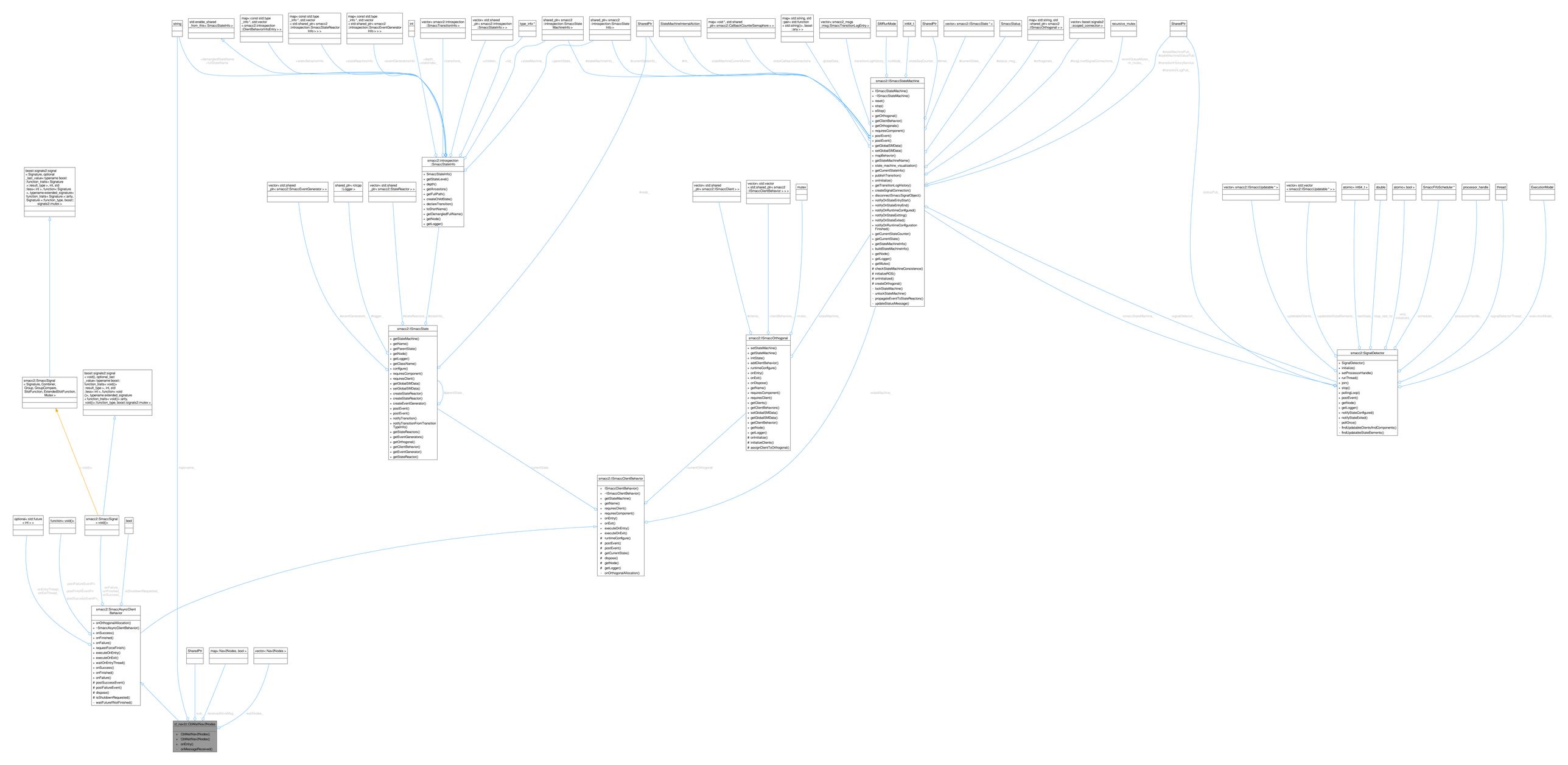
string
std::enable_shared _from_this< SmaccStateInfo >
map< const std::type _info *, std::vector < smacc2::introspection ::ClientBehaviorInfoEntry > >
map< const std::type _info *, std::vector < std::shared_ptr< smacc2 ::introspection::SmaccStateReactor Info > > >
map< const std::type _info *, std::vector < std::shared_ptr< smacc2 ::introspection::SmaccEventGenerator Info > > >
int
vector< smacc2::introspection ::SmaccTransitionInfo >
vector< std::shared _ptr< smacc2::introspection ::SmaccStateInfo > >
type_info *
shared_ptr< smacc2 ::introspection::SmaccState MachineInfo >
shared_ptr< smacc2 ::introspection::SmaccState Info >
SharedPtr
StateMachineInternalAction
map< void *, std::shared _ptr< smacc2::CallbackCounterSemaphore > >
map< std::string, std ::pair< std::function < std::string()>, boost ::array < ...
vector< smacc2::_msgs ::msg::SmaccTransitionLogEntry >
SMRunMode
int64_t
SharedPtr
vector< smacc2::ISmaccState* >
SmaccStatus
map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > >
vector< boost::signals2 ::scoped_connection >
recursive_mutex
SharedPtr
~demangledStateName ~fullStateName
~depth ~stateIndex
~transitions
~children
~rtd
~stateMachine
~parentState
#stateMachineInfo
#currentStateInfo
#rtn
~stateMachineCurrentAction
~stateCallbackConnections
~globalData
~transitionLogHistory
~runMode
~stateSeqCounter
#rtmw
#currentState
#status_msg
~orthogonals
#longLivedSignalConnections
~eventQueueMutex ~m_mutex
#stateMachinePub ~#stateMachineStatusPub
#transitionHistoryService
#transitionLogPub
smacc2::introspection ::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
+ getNode()
+ getLogger()
smacc2::ISmaccStateMachine
+ ISmaccStateMachine()
~ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getClientBehavior()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ 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()
+ getLogger()
+ getMutex()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
~statusPub
vector< smacc2::ISmaccUpdatable * >
vector< std::vector < smacc2::ISmaccUpdatable * > >
atomic< int64_t > ~lastState
double ~loop_rate_hz
atomic< bool > ~end_initialized
SmaccFloSchedule* ~scheduler
processor_handle ~processorHandle
thread ~signalDetectorThread
ExecutionModel ~executionModel
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 >
vector< std::shared _ptr< smacc2::SmaccEventGenerator > >
shared_ptr< rclcpp ::Logger >
vector< std::shared _ptr< smacc2::IStateReactor > >
~logger
~eventGenerators
~stateReactors
~stateInfo
~node
vector< std::shared _ptr< smacc2::ISmaccClient > >
vector< std::vector < std::shared_ptr< smacc2 ::ISmaccClientBehavior > > >
mutex
~clients
~clientBehaviors
~mutex
~stateMachine
smacc2::ISmaccState
+ getStateMachine()
+ getName()
+ getParentState()
+ getNode()
+ getLogger()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getOrthogonal()
+ getClientBehavior()
+ getEventGenerator()
+ getStateReactor()
~parentState
smacc2::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
# onInitialize()
# initializeClients()
# assignClientToOrthogonal()
~stateMachine
smacc2::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
+ pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
~signalDetector
~updatableClients
~updatableStateElements
smacc2::Smacc Signal < 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 >
~void()>
~topicname
smacc2::ISmaccClientBehavior
+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ onEntry()
+ onExit()
+ executeOnEntry()
+ executeOnExit()
# runtimeConfigure()
# postEvent()
# postEvent()
# getCurrentState()
# dispose()
# getNode()
# getLogger()
- onOrthogonalAllocation()
~currentState
~currentOrthogonal
optional< std::future < int > >
function< void()>
smacc2::Smacc Signal < void()>
bool
~onEntryThread ~onExitThread
~postFailureEventFn ~postFinishEventFn ~postSuccessEventFn
~onFailure ~onFinished ~onSuccess
~isShutdownRequested
smacc2::SmaccAsyncClient Behavior
+ onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
+ requestForceFinish()
+ executeOnEntry()
+ executeOnExit()
+ waitOnEntryThread()
+ onSuccess()
+ onFinished()
+ onFailure()
# postSuccessEvent()
# postFailureEvent()
# dispose()
# isShutdownRequested()
- waitFutureIfNotFinished()
SharedPtr
map< Nav2Nodes, bool >
vector< Nav2Nodes >
~sub
~receivedAliveMsg
~waitNodes
cl_nav2::ClWaitNav2Node
+ ClWaitNav2Node()
+ ~ClWaitNav2Node()
+ onEntry()
- onMessageReceived()