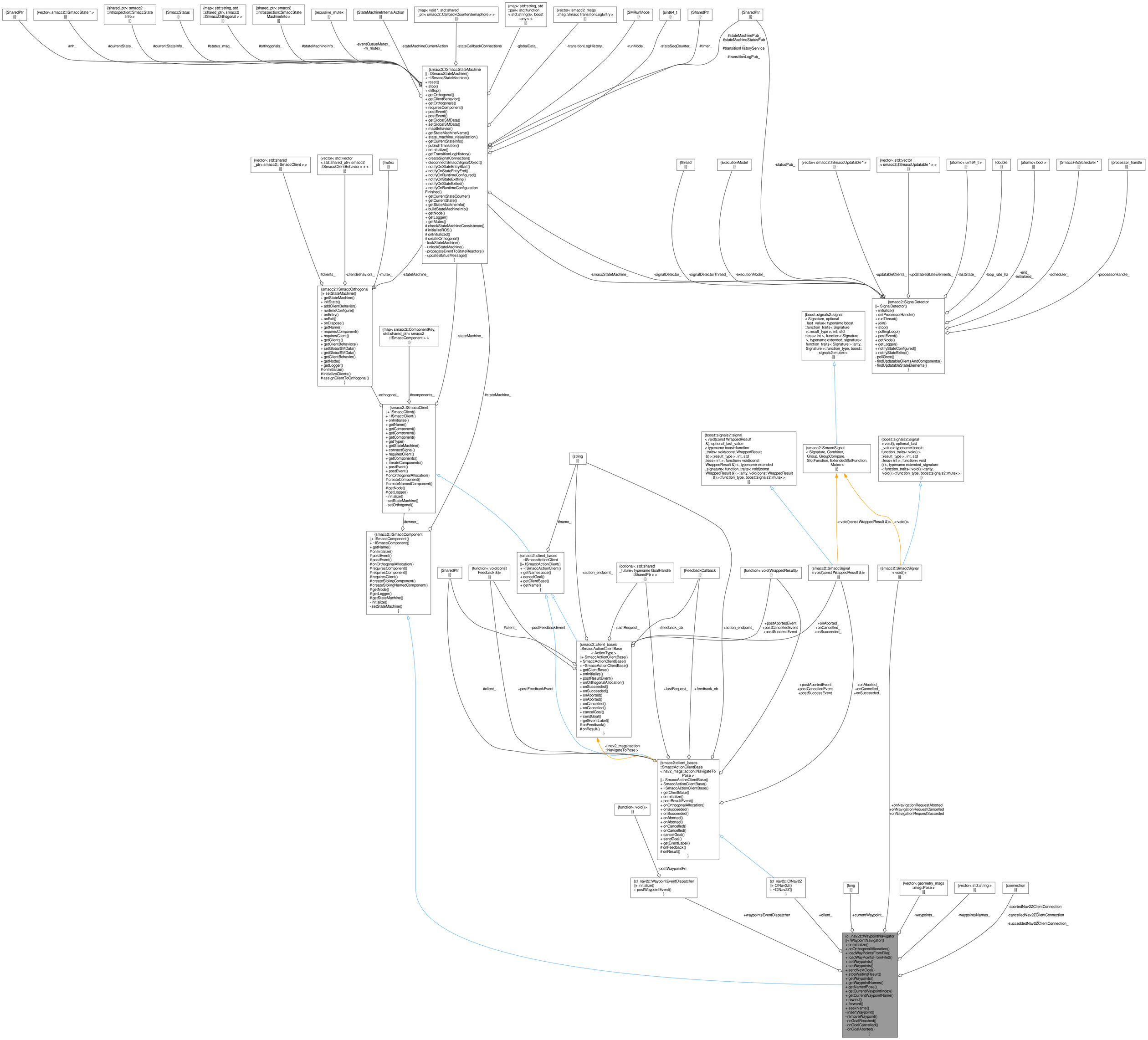
{SharedPtr |||}
{vector< smacc2::ISmaccState * > |||}
{shared_ptr< smacc2 ::introspection::SmaccState Info > |||}
{SmaccStatus |||}
{map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > > |||}
{shared_ptr< smacc2 ::introspection::SmaccState MachineInfo > |||}
{recursive_mutex |||}
{StateMachineInternalAction |||}
{map< void *, std::shared _ptr< smacc2::CallbackCounterSemaphore > > |||}
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > |||}
{vector< smacc2_msgs ::msg::SmaccTransitionLogEntry > |||}
{SMRunMode |||}
{uint64_t |||}
{SharedPtr |||}
{SharedPtr |||}
#nh_
#currentState_
#currentStateInfo_
#status_msg_
#orthogonals_
#stateMachineInfo_
-eventQueueMutex_ -m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#timer_
#stateMachinePub_ #stateMachineStatusPub
#transitionHistoryService
#transitionLogPub_
{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() # checkStateMachineConsistency() # initializeROS() # onInitialized() # createOrthogonal() - lockStateMachine() - unlockStateMachine() - propagateEventToStateReactors() - updateStatusMessage() }
{vector< std::shared _ptr< smacc2::ISmaccClient > > |||}
{vector< std::vector < std::shared_ptr< smacc2 ::ISmaccClientBehavior > > > |||}
{mutex |||}
{thread |||}
{ExecutionModel |||}
-statusPub_
{vector< smacc2::ISmaccUpdatable * > |||}
{vector< std::vector < smacc2::ISmaccUpdatable * > > |||}
{atomic< uint64_t > |||}
{double |||}
{atomic< bool > |||}
{SmaccFifoScheduler * |||}
{processor_handle |||}
#clients_
-clientBehaviors_
-mutex_
-stateMachine_
-smaccStateMachine_
-signalDetector_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
-loop_rate_hz
-end_ -initialized_
-scheduler_
-processorHandle_
{smacc2::ISmaccOrthogonal ||+ setStateMachine() + getStateMachine() + initState() + addClientBehavior() + runtimeConfigure() + onEntry() + onExit() + onDispose() + getName() + requiresComponent() + requiresClient() + getClients() + getClientBehaviors() + setGlobalSMData() + getGlobalSMData() + getClientBehavior() + getNode() + getLogger() # onInitialize() # initializeClients() # assignClientToOrthogonal() }
{smacc2::SignalDetector ||+ SignalDetector() + initialize() + setProcessorHandle() + runThread() + join() + stop() + pollingLoop() + postEvent() + getNode() + getLogger() + notifyStateConfigured() + notifyStateExited() - pollOnce() - findUpdatableClientsAndComponents() - 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 > |||}
{map< smacc2::ComponentKey, std::shared_ptr< smacc2 ::ISmaccComponent > > |||}
-stateMachine_
-orthogonal_
#components_
#stateMachine_
{smacc2::ISmaccClient ||+ ISmaccClient() + ~ISmaccClient() + onInitialize() + getName() + getComponent() + getComponent() + getComponent() + getType() + getStateMachine() + connectSignal() + requiresClient() + getComponents() + iterateComponents() + postEvent() + postEvent() # onOrthogonalAllocation() # createComponent() # createNamedComponent() # getNode() # getLogger() - initialize() - setStateMachine() - setOrthogonal() }
#owner_
{smacc2::ISmaccComponent ||+ ISmaccComponent() + ~ISmaccComponent() + getName() # onInitialize() # postEvent() # postEvent() # onOrthogonalAllocation() # requiresComponent() # requiresComponent() # requiresClient() # createSiblingComponent() # createSiblingNamedComponent() # getNode() # getLogger() # getStateMachine() - initialize() - setStateMachine() }
{boost::signals2::signal < void(const WrappedResult &), optional_last_value < typename boost::function _traits< void(const WrappedResult &) >::result_type >, int, std ::less< int >, function< void(const WrappedResult &) >, typename extended _signature< function_traits< void(const WrappedResult &) >::arity, void(const WrappedResult &) >::function_type, boost::signals2::mutex > |||}
{smacc2::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 > |||}
{string |||}
#name_
< void(const WrappedResult &)>
< void()>
{smacc2::client_bases ::ISmaccActionClient ||+ ISmaccActionClient() + ~ISmaccActionClient() + getNamespace() + cancelGoal() + getClientBase() + getName() }
{SharedPtr |||}
{function< void(const Feedback &)> |||}
+action_endpoint_
{optional< std::shared _future< typename GoalHandle ::SharedPtr > > |||}
{FeedbackCallback |||}
{function< void(WrappedResult)> |||}
{smacc2::SmaccSignal < void(const WrappedResult &)> |||}
{smacc2::SmaccSignal < void()> |||}
#client_
+postFeedbackEvent
+lastRequest_
+feedback_cb
+action_endpoint_
+postAbortedEvent +postCancelledEvent +postSuccessEvent
+onAborted_ +onCancelled_ +onSucceeded_
{smacc2::client_bases ::SmaccActionClientBase < ActionType > ||+ SmaccActionClientBase() + SmaccActionClientBase() + ~SmaccActionClientBase() + getClientBase() + onInitialize() + postResultEvent() + onOrthogonalAllocation() + onSucceeded() + onSucceeded() + onAborted() + onAborted() + onCancelled() + onCancelled() + cancelGoal() + sendGoal() + getEventLabel() # onFeedback() # onResult() }
< nav2_msgs::action ::NavigateToPose >
{smacc2::client_bases ::SmaccActionClientBase < nav2_msgs::action::NavigateTo Pose > ||+ SmaccActionClientBase() + SmaccActionClientBase() + ~SmaccActionClientBase() + getClientBase() + onInitialize() + postResultEvent() + onOrthogonalAllocation() + onSucceeded() + onSucceeded() + onAborted() + onAborted() + onCancelled() + onCancelled() + cancelGoal() + sendGoal() + getEventLabel() # onFeedback() # onResult() }
+onNavigationRequestAborted +onNavigationRequestCancelled +onNavigationRequestSucceeded
{function< void()> |||}
-postWaypointFn
{cl_nav2z::WaypointEventDispatcher ||+ initialize() + postWaypointEvent() }
{cl_nav2z::ClNav2Z ||+ ClNav2Z() + ~ClNav2Z() }
{long |||}
{vector< geometry_msgs ::msg::Pose > |||}
{vector< std::string > |||}
{connection |||}
-abortedNav2ZClientConnection
-cancelledNav2ZClientConnection
-succeddedNav2ZClientConnection_
+waypointsEventDispatcher
+client_
+currentWaypoint_
-waypoints_
-waypointsNames_
{cl_nav2z::WaypointNavigator ||+ WaypointNavigator() + onInitialize() + onOrthogonalAllocation() + loadWayPointsFromFile() + loadWayPointsFromFile2() + setWaypoints() + setWaypoints() + sendNextGoal() + stopWaitingResult() + getWaypoints() + getWaypointNames() + getNamedPose() + getCurrentWaypointIndex() + getCurrentWaypointName() + rewind() + forward() + seekName() - insertWaypoint() - removeWaypoint() - onGoalReached() - onGoalCancelled() - onGoalAborted() }