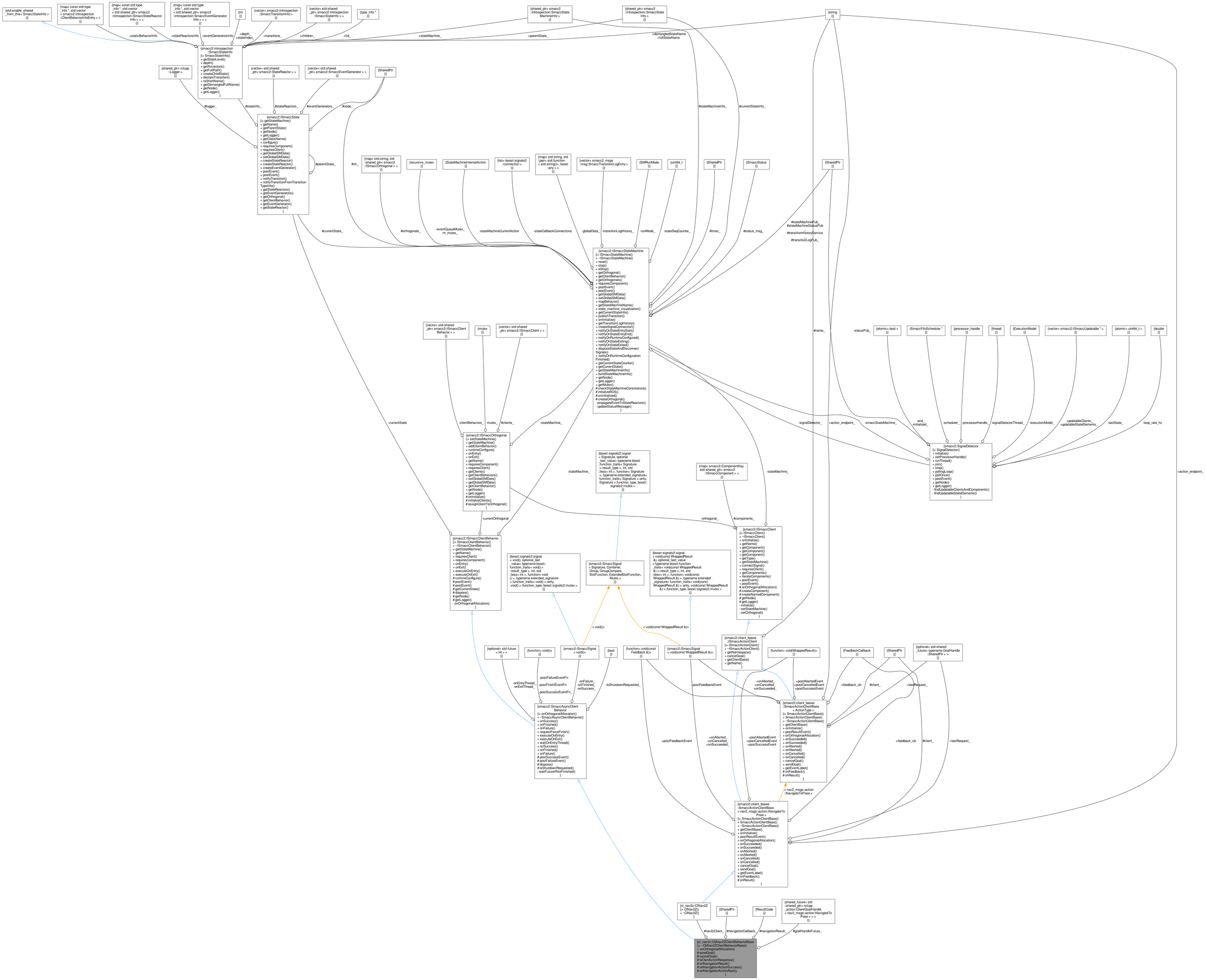
{std::enable_shared
_from_this< SmaccStateInfo >
}}
{map< const std::type
_info *, std::vector
< smacc2::introspection
::ClientBehaviorInfoEntry > >
}}
{map< const std::type
_info *, std::vector
< std::shared_ptr< smacc2
::introspection::SmaccStateReactor
Info > > >
}}
{map< const std::type
_info *, 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 >
}}
{string
}}
+staticBehaviorInfo
+stateReactorsInfo
+eventGeneratorsInfo
+depth_
+stateIndex_
+transitions_
+children_
+sid_
+stateMachine_
+parentState_
+demangledStateName
+fullStateName
{smacc2::introspection
::SmaccStateInfo
|| + SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
+ getNode()
+ getLogger()
}
{shared_ptr< rclcpp
::Logger >
}}
{vector< std::shared
_ptr< smacc2::StateReactor > >
}}
{vector< std::shared
_ptr< smacc2::SmaccEventGenerator > >
}}
{SharedPtr
}}
#logger_
#stateInfo_
#stateReactors_
#eventGenerators_
#node_
#stateMachineInfo_
#currentStateInfo_
{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_
#currentState_
#inh_
{map< std::string, std
::shared_ptr< smacc2
::ISmaccOrthogonal > >
}}
{recursive_mutex
}}
{StateMachineInternalAction
}}
{list< boost::signals2
::connection >
}}
{map< std::string, std
::pair< std::function
< std::string()>, boost
::any > >
}}
{vector< smacc2_msgs
::msg::SmaccTransitionLogEntry >
}}
{SMRunMode
}}
{uint64_t
}}
{SharedPtr
}}
{SmaccStatus
}}
{SharedPtr
}}
#stateMachinePub
#stateMachineStatusPub
#transitionHistoryService
#transitionLogPub_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#timer_
#status_msg_
{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()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExiting()
+ notifyOnStateExited()
+ disposeStateAndDisconnect
Signals()
+ notifyOnRuntimeConfiguration
Finished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
+ getLogger()
+ getMutex()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
- propagateEventToStateReactors()
- updateStatusMessage()
}
#name_
-statusPub_
{atomic< bool >
}}
{SmaccFifoScheduler *
}}
{processor_handle
}}
{thread
}}
{ExecutionModel
}}
{vector< smacc2::ISmaccUpdatable * >
}}
{atomic< uint64_t >
}}
{double
}}
-currentState
{vector< std::shared
_ptr< smacc2::ISmaccClient
Behavior > >
}}
{mutex
}}
{vector< std::shared
_ptr< smacc2::ISmaccClient > >
}}
-clientBehaviors_
-mutex_
#clients_
-stateMachine_
-signalDetector_
+action_endpoint_
-smaccStateMachine_
-end_
-initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
-loop_rate_hz
{smacc2::ISmaccOrthogonal
|| + setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
# onInitialize()
# initializeClients()
# assignClientToOrthogonal()
}
{smacc2::SignalDetector
|| + SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
+ getNode()
+ getLogger()
- 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_
-stateMachine_
-orthogonal_
#components_
-currentOrthogonal
{smacc2::ISmaccClientBehavior
|| + ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ onEntry()
+ onExit()
+ executeOnEntry()
+ executeOnExit()
# runtimeConfigure()
# postEvent()
# postEvent()
# getCurrentState()
# dispose()
# getNode()
# getLogger()
- onOrthogonalAllocation()
}
{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 >
}}
{smacc2::SmaccSignal
< Signature, Combiner,
Group, GroupCompare,
SlotFunction, ExtendedSlotFunction,
Mutex >
}}
{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::ISmaccClient
|| + ISmaccClient()
+ ~ISmaccClient()
+ onInitialize()
+ getName()
+ getComponent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
+ iterateComponents()
+ postEvent()
+ postEvent()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# getNode()
# getLogger()
- initialize()
- setStateMachine()
- setOrthogonal()
}
< void()>
< void(const WrappedResult &)>
{optional< std::future
< int > >
}}
{function< void()>
}}
{smacc2::SmaccSignal
< void()>
}}
{bool
}}
{function< void(const
Feedback &)>
}}
{smacc2::SmaccSignal
< void(const WrappedResult &)>
}}
{smacc2::client_bases
::SmaccActionClient
|| + ISmaccActionClient()
+ ~ISmaccActionClient()
+ getNamespace()
+ cancelGoal()
+ getClientBase()
+ getName()
}
{function< void(WrappedResult)>
}}
{FeedbackCallback
}}
{SharedPtr
}}
{optional< std::shared
_future< typename GoalHandle
::SharedPtr > >
}}
-onEntryThread_
-onExitThread_
-postFailureEventFn
-postFinishEventFn
-postSuccessEventFn_
-onFailure_
-onFinished_
-onSuccess_
-isShutdownRequested_
+postFeedbackEvent
+onAborted_
+onCancelled_
+onSucceeded_
+postAbortedEvent
+postCancelledEvent
+postSuccessEvent
+feedback_cb
#client_
+lastRequest_
{smacc2::SmaccAsyncClient
Behavior
|| + onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
+ requestForceFinish()
+ executeOnEntry()
+ executeOnExit()
+ waitOnEntryThread()
+ onSuccess()
+ onFinished()
+ onFailure()
# postSuccessEvent()
# postFailureEvent()
# dispose()
# isShutdownRequested()
- waitFutureIfNotFinished()
}
{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()
}
{cl_nav2z::ClNav2Z
|| + ClNav2Z()
+ ~ClNav2Z()
}
{SharedPtr
}}
{ResultCode
}}
{shared_future< std
::shared_ptr< rclcpp
_action::ClientGoalHandle
< nav2_msgs::action::NavigateTo
Pose > > >
}}
{cl_nav2z::ClNav2ZClientBehaviorBase
|| + ~ClNav2ZClientBehaviorBase()
+ onOrthogonalAllocation()
# sendGoal()
# cancelGoal()
# isOwnActionResponse()
# onNavigationResult()
# onNavigationActionSuccess()
# onNavigationActionAbort()
}
#nav2zClient_
#navigationCallback_
#navigationResult_
#goalHandleFuture_