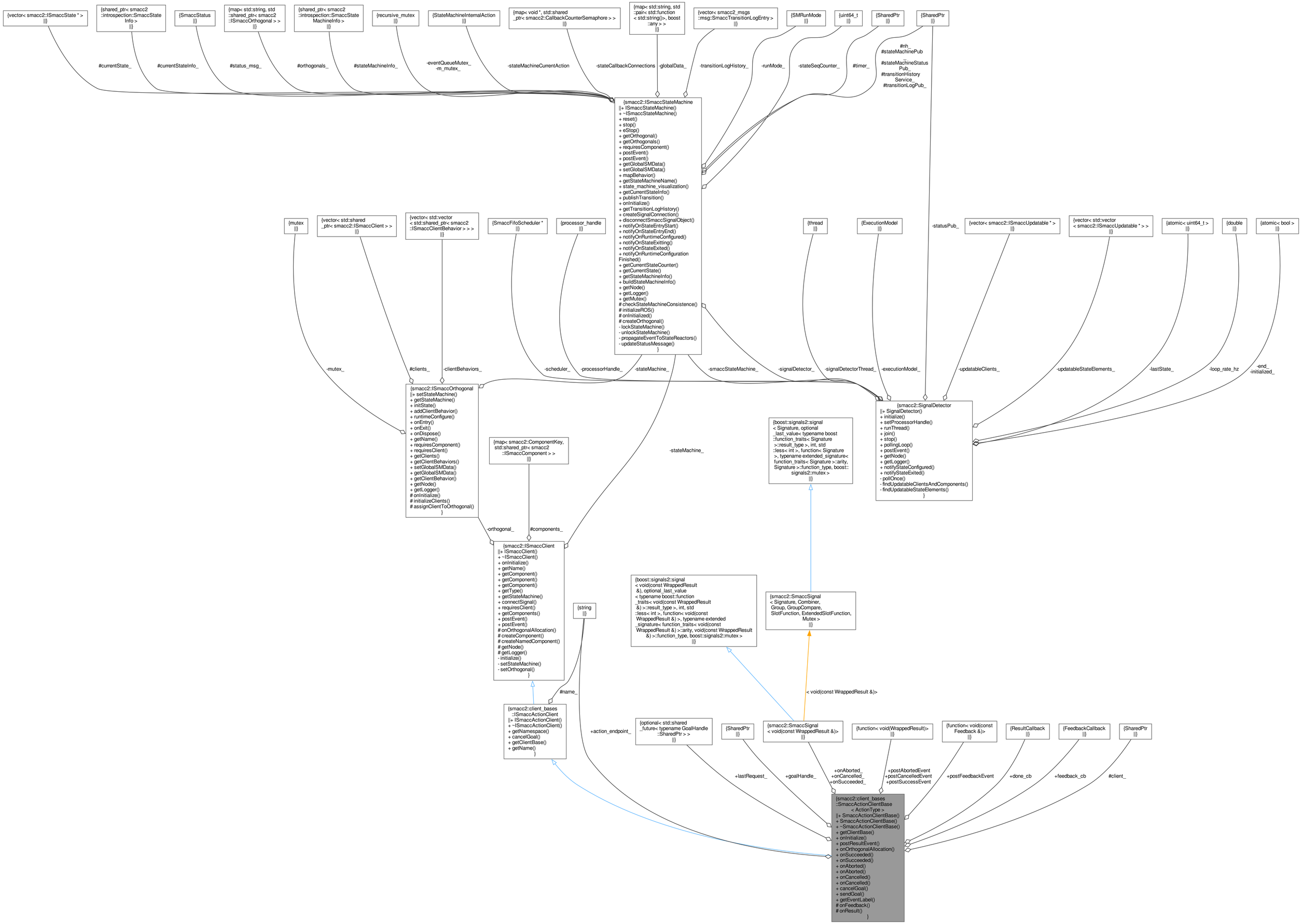
{vector< smacc2::ISmaccState * > }|}
{shared_ptr< smacc2 ::introspection::SmaccState Info > }|}
{SmaccStatus }|}
{map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > > }|}
{shared_ptr< smacc2 ::introspection::SmaccState MachineInfo > }|}
{recursive_mutex }|}
{StateMachineInternalAction }|}
{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 }|}
#currentState_
#currentStateInfo_
#status_msg_
#orthogonals_
#stateMachineInfo_
-eventQueueMutex_ -m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#timer_
#nh_ #stateMachinePub_ #stateMachineStatus Pub_ #transitionHistory Service_ #transitionLogPub_
{smacc2::ISmaccStateMachine }|+ ISmaccStateMachine() + ~ISmaccStateMachine() + reset() + stop() + eStop() + getOrthogonal() + 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() }
{mutex }|}
{vector< std::shared _ptr< smacc2::ISmaccClient > > }|}
{vector< std::vector < std::shared_ptr< smacc2 ::ISmaccClientBehavior > > > }|}
{SmaccFifoScheduler * }|}
{processor_handle }|}
{thread }|}
{ExecutionModel }|}
-statusPub_
{vector< smacc2::ISmaccUpdatable * > }|}
{vector< std::vector < smacc2::ISmaccUpdatable * > > }|}
{atomic< uint64_t > }|}
{double }|}
{atomic< bool > }|}
-mutex_
#clients_
-clientBehaviors_
-scheduler_
-processorHandle_
-stateMachine_
-smaccStateMachine_
-signalDetector_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
-loop_rate_hz
-end_ -initialized_
{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() }
{map< smacc2::ComponentKey, std::shared_ptr< smacc2 ::ISmaccComponent > > }|}
-stateMachine_
{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 > }|}
-orthogonal_
#components_
{smacc2::ISmaccClient }|+ ISmaccClient() + ~ISmaccClient() + onInitialize() + getName() + getComponent() + getComponent() + getComponent() + getType() + getStateMachine() + connectSignal() + requiresClient() + getComponents() + postEvent() + postEvent() # onOrthogonalAllocation() # createComponent() # createNamedComponent() # getNode() # getLogger() - initialize() - setStateMachine() - setOrthogonal() }
{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 > }|}
{string }|}
#name_
< void(const WrappedResult &)>
{smacc2::client_bases ::ISmaccActionClient }|+ ISmaccActionClient() + ~ISmaccActionClient() + getNamespace() + cancelGoal() + getClientBase() + getName() }
+action_endpoint_
{optional< std::shared _future< typename GoalHandle ::SharedPtr > > }|}
{SharedPtr }|}
{smacc2::SmaccSignal < void(const WrappedResult &)> }|}
{function< void(WrappedResult)> }|}
{function< void(const Feedback &)> }|}
{ResultCallback }|}
{FeedbackCallback }|}
{SharedPtr }|}
+lastRequest_
+goalHandle_
+onAborted_ +onCancelled_ +onSucceeded_
+postAbortedEvent +postCancelledEvent +postSuccessEvent
+postFeedbackEvent
+done_cb
+feedback_cb
#client_
{smacc2::client_bases ::SmaccActionClientBase < ActionType > }|+ SmaccActionClientBase() + SmaccActionClientBase() + ~SmaccActionClientBase() + getClientBase() + onInitialize() + postResultEvent() + onOrthogonalAllocation() + onSucceeded() + onSucceeded() + onAborted() + onAborted() + onCancelled() + onCancelled() + cancelGoal() + sendGoal() + getEventLabel() # onFeedback() # onResult() }