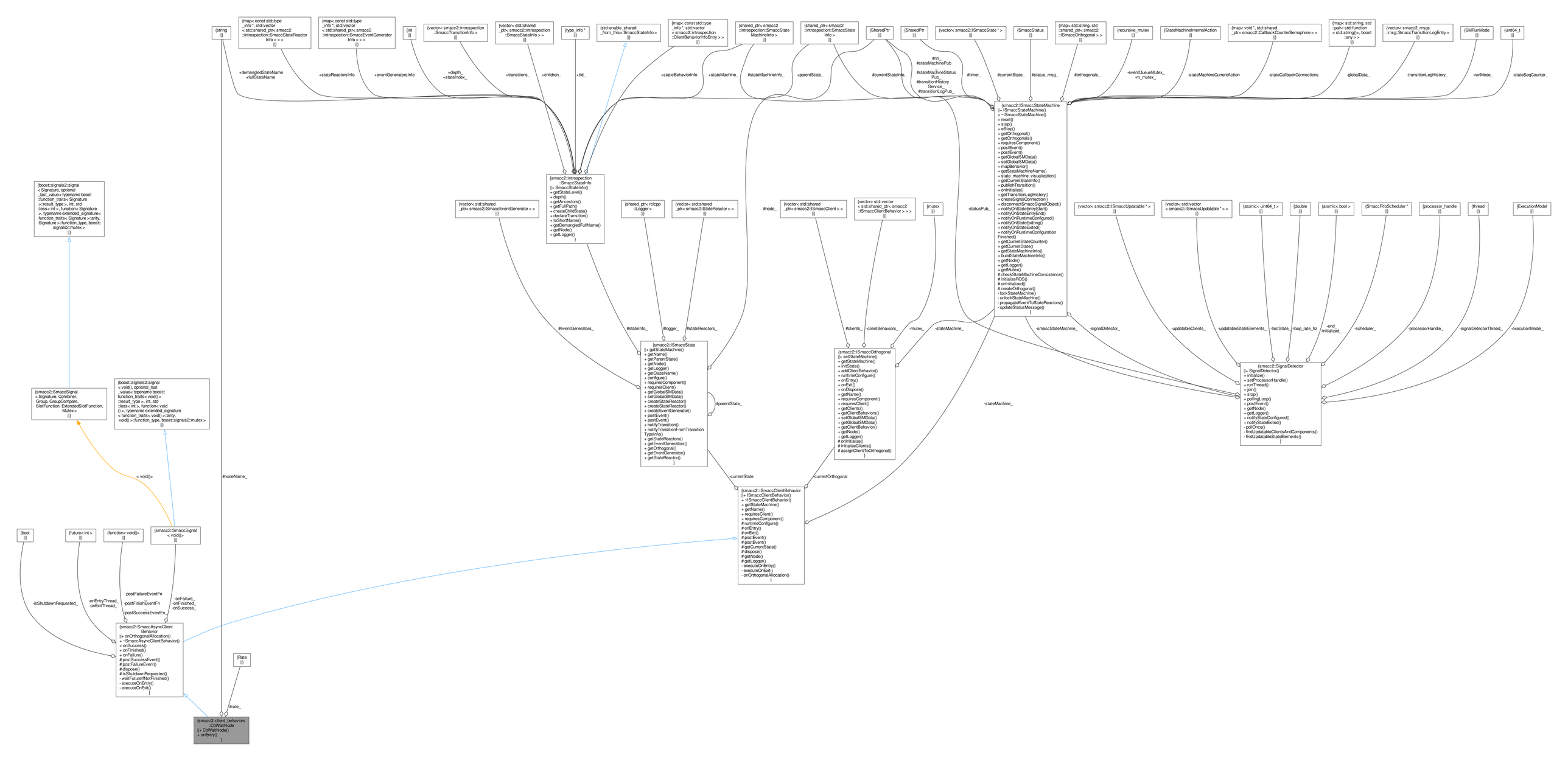
{string}
{map< const std::type_info *, std::vector< std::shared_ptr< smacc2::introspection::SmaccStateReactorInfo > > >}
{map< const std::type_info *, std::vector< std::shared_ptr< smacc2::introspection::SmaccEventGeneratorInfo > > >}
{int}
{vector< smacc2::introspection::SmaccTransitionInfo >}
{vector< std::shared_ptr< smacc2::introspection::SmaccStateInfo > >}
{type_info *}
{std::enable_shared_from_this< SmaccStateInfo >}
{map< const std::type_info *, std::vector< smacc2::introspection::ClientBehaviorInfoEntry > >}
{shared_ptr< smacc2::introspection::SmaccStateMachineInfo >}
{shared_ptr< smacc2::introspection::SmaccStateInfo >}
{SharedPtr}
{SharedPtr}
{vector< smacc2::ISmaccState * >}
{SmaccStatus}
{map< std::string, std::shared_ptr< smacc2::ISmaccOrthogonal > >}
{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}
+demangledStateName
+fullStateName
+stateReactorsInfo
+eventGeneratorsInfo
+depth_
+stateIndex_
+transitions_
+children_
+tid_
+staticBehaviorInfo
+stateMachine_
#stateMachineInfo_
+parentState_
#currentStateInfo_
#nh_
#stateMachinePub
#stateMachineStatusPub_
#transitionHistoryService_
#transitionLogPub_
#timer_
#currentState_
#status_msg_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
{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()
}
{smacc2::introspection::SmaccStateInfo
|+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
+ getNode()
+ getLogger()
}
{vector< std::shared_ptr< smacc2::SmaccEventGenerator > >}
{shared_ptr< rclcpp::Logger >}
{vector< std::shared_ptr< smacc2::StateReactor > >}
#node_
{vector< std::shared_ptr< smacc2::ISmaccClient > >}
{vector< std::vector< std::shared_ptr< smacc2::ISmaccClientBehavior > > >}
{mutex}
-statusPub_
{vector< smacc2::ISmaccUpdatable * >}
{vector< std::vector< smacc2::ISmaccUpdatable * > >}
{atomic< uint64_t >}
{double}
{atomic< bool >}
{SmaccFifoScheduler *}
{processor_handle}
{thread}
{ExecutionModel}
#eventGenerators_
#stateInfo_
#logger_
#stateReactors_
#clients_
-clientBehaviors_
-mutex_
-stateMachine_
-smaccStateMachine_
-signalDetector_
-updatableClients_
-updatableStateElements_
-lastState_
-loop_rate_hz
-end_
-initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
{smacc2::ISmaccState
|+ getStateMachine()
+ getName()
+ getParentState()
+ getNode()
+ getLogger()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getOrthogonal()
+ 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()
}
{smacc2::SignalDetector
|+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
}
-currentState
-currentOrthogonal
-stateMachine_
{smacc2::ISmaccClientBehavior
|+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# getCurrentState()
# dispose()
# getNode()
# getLogger()
- executeOnEntry()
- executeOnExit()
- onOrthogonalAllocation()
}
{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 >
}
{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 >
}
< void()>
{smacc2::SmaccSignal
< void()>
}
#nodeName_
{bool}
{future< int >}
{function< void()>}
-isShutdownRequested_
-onEntryThread_
-onExitThread_
-postFailureEventFn
-postFinishEventFn
-postSuccessEventFn
-onFailure_
-onFinished_
-onSuccess_
{smacc2::SmaccAsyncClient Behavior
|+ onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
# postSuccessEvent()
# postFailureEvent()
# dispose()
# isShutdownRequested()
- waitFutureIfNotFinished()
- executeOnEntry()
- executeOnExit()
}
{Rate}
#rate_
{smacc2::client_behaviors
::CbWaitNode
|+ CbWaitNode()
+ onEntry()
}