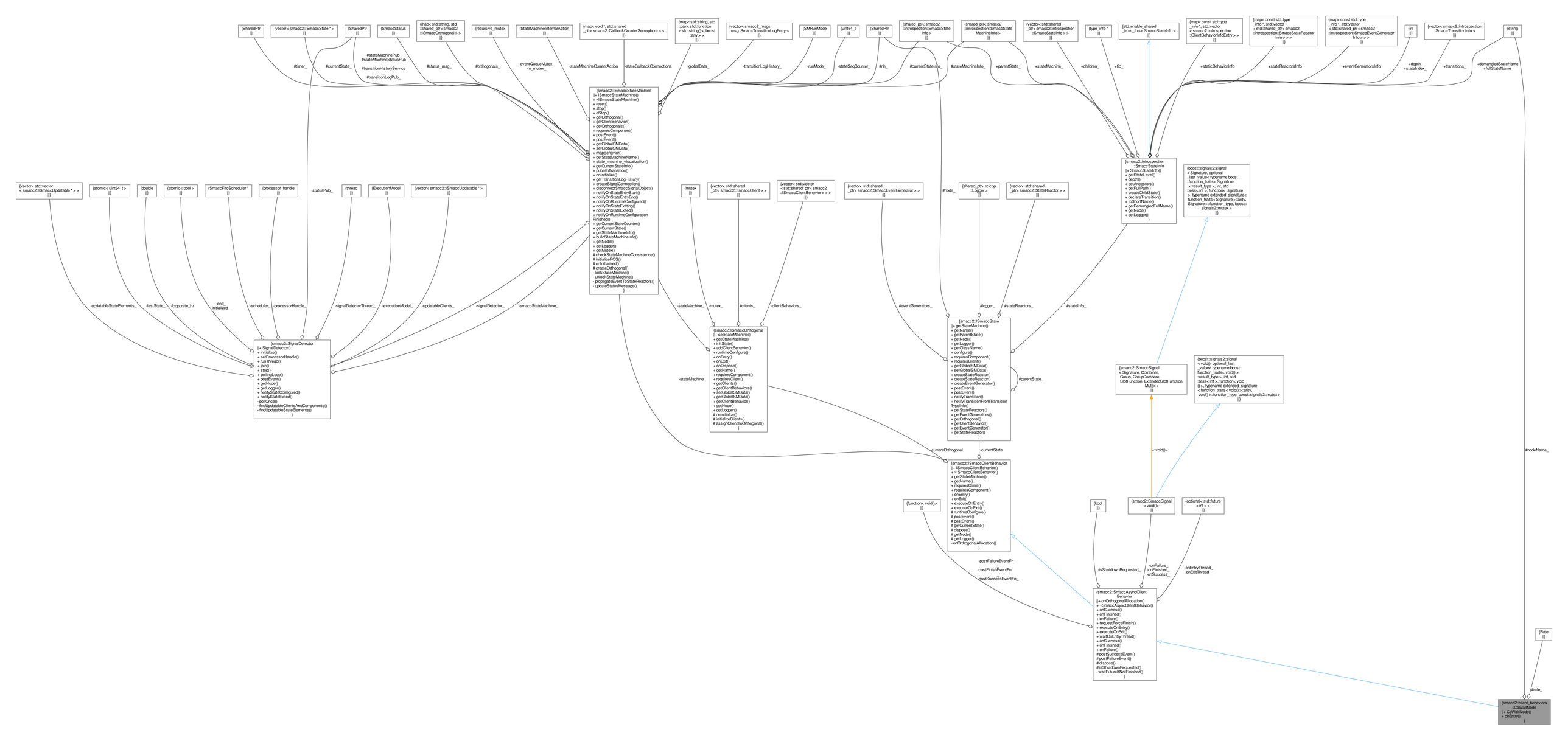
{SharedPtr }
{vector< smacc2::ISmaccState * > }
{SharedPtr }
{SmaccStatus }
{map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > > }
{recursive_mutex }
{StateMachineInternalAction }
{map< void *, std::shared _ptr< smacc2::CallbackCounterSemaphore > > }
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > }
{vector< smacc2_msgs ::msg::SmaccTransitionLogEntry > }
{SMRunMode }
{uint64_t }
{SharedPtr }
{shared_ptr< smacc2 ::introspection::SmaccState Info > }
{shared_ptr< smacc2 ::introspection::SmaccState MachineInfo > }
{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 > > }
{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 > }
{string }
#timer_
#currentState_
#stateMachinePub_
#stateMachineStatusPub_
#transitionHistoryService_
#transitionLogPub_
#status_msg_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#n_
#currentStateInfo_
#stateMachineInfo_
+parentState_
+stateMachine_
+children_
+tid_
+staticBehaviorInfo
+stateReactorsInfo
+eventGeneratorsInfo
+depth_
+stateIndex_
+transitions_
+demangledStateName
+fullStateName
{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()
+ fullstateMachineInfo()
+ 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()
}
{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::vector
< smacc2::ISmaccUpdatable * > >
}
{atomic< uint64_t >
}
{double
}
{atomic< bool >
}
{SmaccFifoScheduler *
}
{processor_handle
}
-statusPub_
{thread
}
{ExecutionModel
}
{vector< smacc2::ISmaccUpdatable * >
}
{mutex
}
{vector< std::shared
_ptr< smacc2::ISmaccClient > >
}
{vector< std::vector
< std::shared_ptr< smacc2
::ISmaccClientBehavior > > >
}
{vector< std::shared
_ptr< smacc2::ISmaccEventGenerator > >
}
#node_
{shared_ptr< rclcpp
::Logger >
}
{vector< std::shared
_ptr< smacc2::StateReactor > >
}
-updatableStateElements_
-lastState_
-loop_rate_hz
-end_
-initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-signalDetector_
-smaccStateMachine_
-stateMachine_
-mutex_
#clients_
-clientBehaviors_
#eventGenerators_
#logger_
#stateReactors_
#stateInfo_
{smacc2::SignalDetector
|+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
}
{smacc2::ISmaccOrthogonal
|+ setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
#onInitialize()
#initializeClients()
#assignClientToOrthogonal()
}
{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_
-stateMachine_
-currentOrthogonal
-currentState
{smacc2::ISmaccClientBehavior
|+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ onEntry()
+ onExit()
+ executeOnEntry()
+ executeOnExit()
#runtimeConfigure()
#postEvent()
#postEvent()
#getCurrentState()
#dispose()
#getNode()
#getLogger()
- onOrthogonalAllocation()
}
{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()>
{function< void()>
}
{bool
}
{smacc2::Smacc Signal
< void()>
}}
{optional< std::future
< int > >
}}
-postFailureEventFn
-postFinishEventFn
-postSuccessEventFn_
-isShutdownRequested_
-onFailure_
-onFinished_
-onSuccess_
-onEntryThread_
-onExitThread_
{smacc2::SmaccAsyncClient Behavior
|+ onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
+ requestIfForceFinish()
+ executeOnEntry()
+ executeOnExit()
+ waitOnEntryThread()
+ onSuccess()
+ onFinished()
+ onFailure()
#postSuccessEvent()
#postFailureEvent()
#dispose()
#isShutdownRequested()
- waitFutureIfNotFinished()
}
#nodeName_
{Rate
}}
#rate_
{smacc2::client_behaviors ::CbWaitNode
|+ CbWaitNode()
+ onEntry()
}