

smacc::ISmaccStateMachine

nh_
 # private_nh_
 # timer_
 # stateMachinePub_
 # stateMachineStatusPub_
 # transitionLogPub_
 # transitionHistoryService_
 # currentState_
 # currentStateInfo_
 # status_msg_
 # orthogonalS_
 - m_mutex_
 - eventQueueMutex_
 - stateMachineCurrentAction_
 - stateCallbackConnections_
 - globalData_
 - transitionLogHistory_
 - runMode_
 - signalDetector_
 - stateSeqCounter_
 - stateMachineInfo_

+ ISmaccStateMachine()
 + ~ISmaccStateMachine()
 + reset()
 + stop()
 + eStop()
 + getOrthogonal()
 + getOrthogonals()
 + requiresComponent()
 + postEvent()
 + postEvent()
 + getTransitionLogHistory()
 + getGlobalSMDData()
 + setGlobalSMDData()
 + mapBehavior()
 + getStateMachineName()
 + state_machine_visualization()
 + getCurrentStateInfo()
 + publishTransition()
 + onInitialize()
 + getTransitionLogHistory()
 + createSignalConnection()
 + notifyOnStateEntryStart()
 + notifyOnStateEntryEnd()
 + notifyOnRuntimeConfigured()
 + notifyOnStateExiting()
 + notifyOnStateExited()
 + notifyOnRuntimeConfigurationFinished()
 + getCurrentStateCounter()
 + getCurrentState()
 + getStateMachineInfo()
 + buildStateMachineInfo()
 # checkStateMachineConsistence()
 # initializeROS()
 # onInitialized()
 # createOrthogonal()
 # getParam()
 # setParam()
 # param()
 - lockStateMachine()
 - unlockStateMachine()
 - propagateEventToStateReactors()
 - updateStatusMessage()

boost::statechart::

asynchronous_state_machine

< DerivedStateMachine, InitialState
 Type, SmaccFifoScheduler, SmaccAllocator >

smacc::SmaccStateMachine

Base< DerivedStateMachine,

InitialStateType >

+ SmaccStateMachineBase()
 + ~SmaccStateMachineBase()
 + reset()
 + stop()
 + eStop()
 + initiate_impl()

sc::asynchronous_state

_machine< SmCalendarWeek

, MsWorkweek , SmaccFifoScheduler,

SmaccAllocator >

smacc::SmaccStateMachine

Base< SmCalendarWeek,

MsWorkweek >

+ SmaccStateMachineBase()
 + ~SmaccStateMachineBase()
 + reset()
 + stop()
 + eStop()
 + initiate_impl()

sm_calendar_week::SmCalendarWeek

+ onInitialize()

< SmCalendarWeek, MsWorkweek >