

smacc2::ISmaccStateMachine

# nh\_  
 # timer\_  
 # stateMachinePub\_  
 # stateMachineStatusPub\_  
 # transitionLogPub\_  
 # transitionHistoryService\_  
 # currentState\_  
 # currentStateInfo\_  
 # status\_msg\_  
 # orthogonals\_  
 # stateMachineInfo\_  
 - m\_mutex\_  
 - eventQueueMutex\_  
 - stateMachineCurrentAction  
 - stateCallbackConnections  
 - globalData\_  
 - transitionLogHistory\_  
 - runMode\_  
 - signalDetector\_  
 - stateSeqCounter\_

+ ISmaccStateMachine()  
 + ~ISmaccStateMachine()  
 + reset()  
 + stop()  
 + eStop()  
 + getOrthogonal()  
 + getOrthogonals()  
 + requiresComponent()  
 + postEvent()  
 + postEvent()  
 + getGlobalSMDData()  
 + setGlobalSMDData()  
 + 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()

boost::statechart::  
asynchronous\_state\_machine  
< DerivedStateMachine, InitialState  
Type, SmaccFifoScheduler, SmaccAllocator >

smacc2::SmaccStateMachine  
Base< DerivedStateMachine,  
InitialStateType >

+ SmaccStateMachineBase()  
 + ~SmaccStateMachineBase()  
 + reset()  
 + stop()  
 + eStop()  
 + initiate\_impl()

boost::statechart::  
asynchronous\_state\_machine  
< SmRespira1, MsRun, SmaccFifoScheduler,  
SmaccAllocator >

smacc2::SmaccStateMachine  
Base< SmRespira1, MsRun >

+ SmaccStateMachineBase()  
 + ~SmaccStateMachineBase()  
 + reset()  
 + stop()  
 + eStop()  
 + initiate\_impl()

sm\_respira\_1::SmRespira1

+ onInitialize()

< SmRespira1, MsRun >