

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::SmaccStateMachineBase< DerivedStateMachine, InitialStateType >

+ SmaccStateMachineBase()

+ ~SmaccStateMachineBase()

+ reset()

+ stop()

+ eStop()

+ initiate_impl()

boost::statechart::asynchronous_state_machine< SmDanceBot2, MsDanceBotRunMode, SmaccFifoScheduler, SmaccAllocator >

smacc::SmaccStateMachineBase< SmDanceBot2, MsDanceBotRunMode >

+ SmaccStateMachineBase()

+ ~SmaccStateMachineBase()

+ reset()

+ stop()

+ eStop()

+ initiate_impl()

sm_dance_bot_2::SmDanceBot2

+ counter_1

+ rt_ready_flag

+ onInitialize()

< SmDanceBot2, MsDanceBotRunMode >