

```
{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()
+ getClientBehavior()
+ 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
< SmAtomicPerformanceTrace1,
State1, SmaccFifoScheduler,
SmaccAllocator >
||}
```

```
{smacc2::SmaccStateMachine
Base< SmAtomicPerformanceTrace1,
State1 >
||+ SmaccStateMachineBase()
+ ~SmaccStateMachineBase()
+ reset()
+ stop()
+ eStop()
+ initiate_impl()
}
```

```
{sm_atomic_performance
_trace_1::SmAtomicPerformance
Trace1
||+ onInitialize()
}
```

< SmAtomicPerformanceTrace1,  
State1 >

