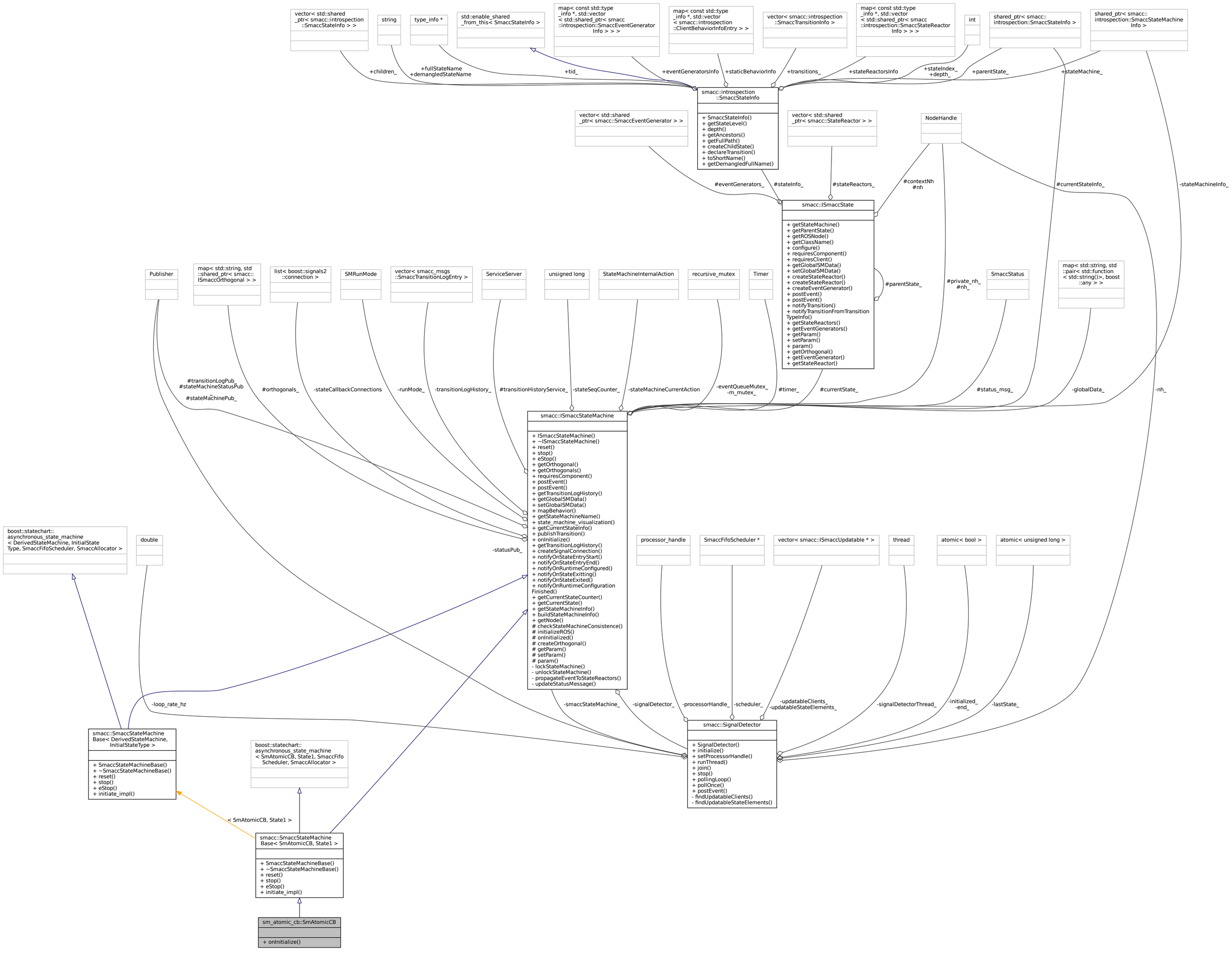
vector< std::shared _ptr< smacc::introspection ::SmaccStateInfo > >
string
type_info *
std::enable_shared _from_this< SmaccStateInfo >
map< const std::type _info *, std::vector < std::shared_ptr< smacc ::introspection::SmaccEventGenerator Info > > >
map< const std::type _info *, std::vector < smacc::introspection ::ClientBehaviorInfoEntry > >
vector< smacc::introspection ::SmaccTransitionInfo >
map< const std::type _info *, std::vector < std::shared_ptr< smacc ::introspection::SmaccStateReactor Info > > >
int
shared_ptr< smacc:: introspection::SmaccStateInfo >
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
+children_
+fullStateName
+demangledStateName
+tid_
+eventGeneratorsInfo
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+stateIndex_
+depth_
+parentState_
+stateMachine_
smacc::introspection ::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
vector< std::shared _ptr< smacc::SmaccEventGenerator > >
vector< std::shared _ptr< smacc::StateReactor > >
NodeHandle
#eventGenerators_
#stateInfo_
#stateReactors_
#contextNh
#nh
#currentStateInfo_
-stateMachineInfo_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getEventGenerator()
+ getStateReactor()
#parentState_
#private_nh_
#nh_
Publisher
map< std::string, std ::shared_ptr< smacc:: ISmaccOrthogonal > >
list< boost::signals2 ::connection >
SMRunMode
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
unsigned long
StateMachineInternalAction
recursive_mutex
Timer
SmaccStatus
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
#transitionLogPub_
#stateMachineStatusPub_
#stateMachinePub_
#orthogonals_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
-stateMachineCurrentAction
-eventQueueMutex_
-m_mutex_
#timer_
#currentState_
#status_msg_
-globalData_
-nh_
smacc::ISmaccStateMachine
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExitting()
+ notifyOnStateExited()
+ notifyOnRuntimeConfiguration Finished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
-statusPub_
boost::statechart:: asynchronous_state_machine < DerivedStateMachine, InitialState Type, SmaccFifoScheduler, SmaccAllocator >
double
processor_handle
SmaccFifoScheduler *
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
-loop_rate_hz
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-updatableClients_
-updatableStateElements_
-signalDetectorThread_
-initialized_
-end_
-lastState_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
smacc::SmaccStateMachine Base< DerivedStateMachine, InitialStateType >
+ SmaccStateMachineBase()
+ ~SmaccStateMachineBase()
+ reset()
+ stop()
+ eStop()
+ initiate_impl()
boost::statechart:: asynchronous_state_machine < SmAtomicCB, State1, SmaccFifo Scheduler, SmaccAllocator >
< SmAtomicCB, State1 >
smacc::SmaccStateMachine Base< SmAtomicCB, State1 >
+ SmaccStateMachineBase()
+ ~SmaccStateMachineBase()
+ reset()
+ stop()
+ eStop()
+ initiate_impl()
sm_atomic_cb::SmAtomicCB
+ onInitialize()