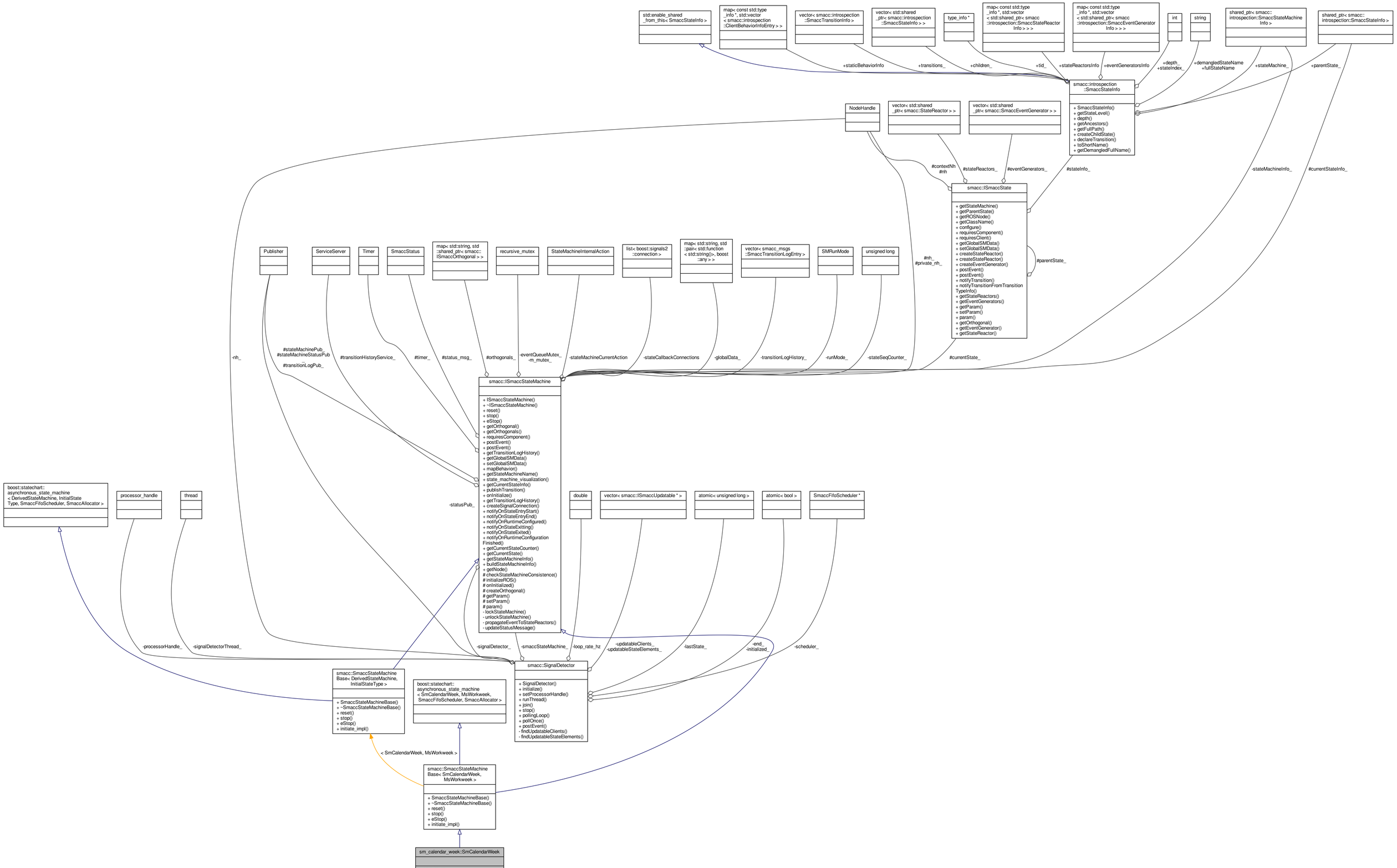
std::enable_shared
_from_this< SmaccStateInfo >
map< const std::type
_info *, std::vector
< smacc::introspection
::ClientBehaviorInfoEntry > >
vector< smacc::introspection
::SmaccTransitionInfo >
vector< std::shared
_ptr< smacc::introspection
::SmaccStateInfo > >
type_info *
map< const std::type
_info *, std::vector
< std::shared_ptr< smacc
::introspection::SmaccStateReactor
Info > > >
map< const std::type
_info *, std::vector
< std::shared_ptr< smacc
::introspection::SmaccEventGenerator
Info > > >
int
string
shared_ptr< smacc::
introspection::SmaccStateMachine
Info >
shared_ptr< smacc::
introspection::SmaccStateInfo >
+staticBehaviorInfo
+transitions_
+children_
+tid_
+stateReactorsInfo
+eventGeneratorsInfo
+depth_
+stateIndex_
+demangledStateName
+fullStateName
+stateMachine_
+parentState_
smacc::introspection
::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
NodeHandle
vector< std::shared
_ptr< smacc::StateReactor > >
vector< std::shared
_ptr< smacc::SmaccEventGenerator > >
#contextNh
#nh
#stateReactors_
#eventGenerators_
#stateInfo_
-stateMachineInfo_
#currentStateInfo_
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_
Publisher
ServiceServer
Timer
SmaccStatus
map< std::string, std
::shared_ptr< smacc::
ISmaccOrthogonal > >
recursive_mutex
StateMachineInternalAction
list< boost::signals2
::connection >
map< std::string, std
::pair< std::function
< std::string()>, boost
::any > >
vector< smacc_msgs
::SmaccTransitionLogEntry >
SMRunMode
unsigned long
#nh_
#private_nh_
-nh_
#stateMachinePub_
#stateMachineStatusPub
_
#transitionLogPub_
#transitionHistoryService_
#timer_
#status_msg_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#currentState_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ 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 >
processor_handle
thread
double
vector< smacc::ISmaccUpdatable * >
atomic< unsigned long >
atomic< bool >
SmaccFifoScheduler *
-processorHandle_
-signalDetectorThread_
-signalDetector_
-smaccStateMachine_
-loop_rate_hz
-updatableClients_
-updatableStateElements_
-lastState_
-end_
-initialized_
-scheduler_
smacc::SmaccStateMachine
Base< DerivedStateMachine,
InitialStateType >
+ SmaccStateMachineBase()
+ ~SmaccStateMachineBase()
+ reset()
+ stop()
+ eStop()
+ initiate_impl()
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
boost::statechart::
asynchronous_state_machine
< SmCalendarWeek, MsWorkweek,
SmaccFifoScheduler, SmaccAllocator >
< SmCalendarWeek, MsWorkweek >
smacc::SmaccStateMachine
Base< SmCalendarWeek,
MsWorkweek >
+ SmaccStateMachineBase()
+ ~SmaccStateMachineBase()
+ reset()
+ stop()
+ eStop()
+ initiate_impl()
sm_calendar_week::SmCalendarWeek
+ onInitialize()