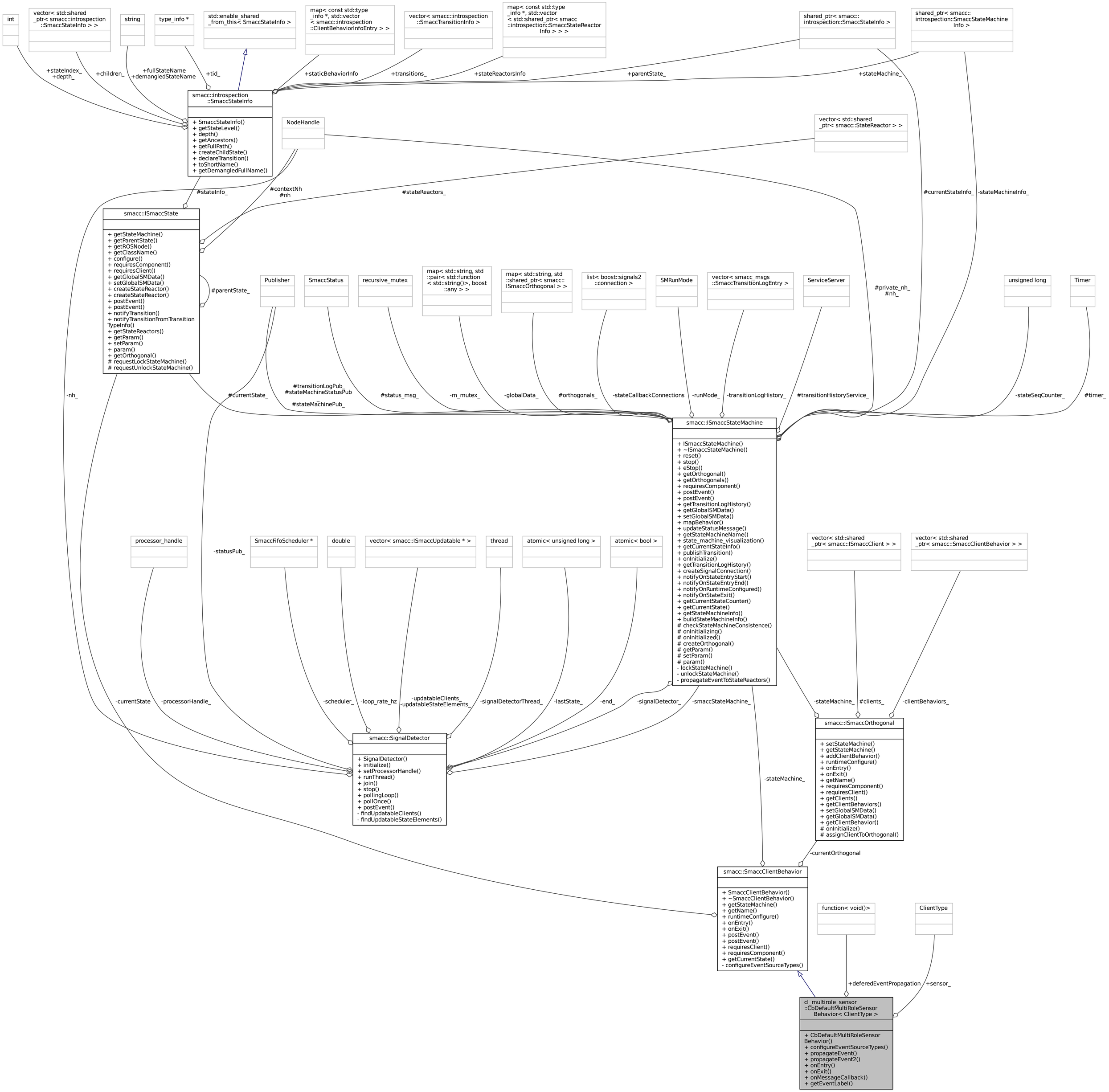
int
vector< std::shared _ptr< smacc::introspection ::SmaccStateInfo > >
string
type_info *
std::enable_shared _from_this< SmaccStateInfo >
map< const std::type _info *, std::vector < smacc::introspection ::ClientBehaviorInfoEntry > >
vector< smacc::introspection ::SmaccTransitionInfo >
map< const std::type _info *, std::vector < std::shared_ptr< smacc ::introspection::SmaccStateReactor Info > > >
shared_ptr< smacc:: introspection::SmaccStateInfo >
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
+stateIndex_ +depth_
+children_
+fullStateName +demangledStateName
+tid_
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+parentState_
+stateMachine_
smacc::introspection ::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
NodeHandle
vector< std::shared _ptr< smacc::StateReactor > >
#stateInfo_
#contextNh #nh
#stateReactors_
#currentStateInfo_
-stateMachineInfo_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateReactors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
# requestLockStateMachine()
# requestUnlockStateMachine()
#parentState_
Publisher
SmaccStatus
recursive_mutex
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
map< std::string, std ::shared_ptr< smacc:: ISmaccOrthogonal > >
list< boost::signals2 ::connection >
SMRunMode
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
#private_nh_ #nh_
unsigned long
Timer
-nh_
#currentState_
#transitionLogPub_ #stateMachineStatusPub_
#stateMachinePub_
#status_msg_
-m_mutex_
-globalData_
#orthogonals_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
#timer_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExit()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
# checkStateMachineConsistence()
# onInitializing()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
processor_handle
-statusPub_
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< unsigned long >
atomic< bool >
vector< std::shared _ptr< smacc::ISmaccClient > >
vector< std::shared _ptr< smacc::SmaccClientBehavior > >
-currentState
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClients_ -updatableStateElements_
-signalDetectorThread_
-lastState_
-end_
-signalDetector_
-smaccStateMachine_
-stateMachine_
#clients_
-clientBehaviors_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
-stateMachine_
smacc::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
-currentOrthogonal
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getStateMachine()
+ getName()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
+ getCurrentState()
- configureEventSourceTypes()
function< void()>
ClientType
+deferedEventPropagation
+sensor_
cl_multirole_sensor ::CbDefaultMultiRoleSensor Behavior< ClientType >
+ CbDefaultMultiRoleSensor Behavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()