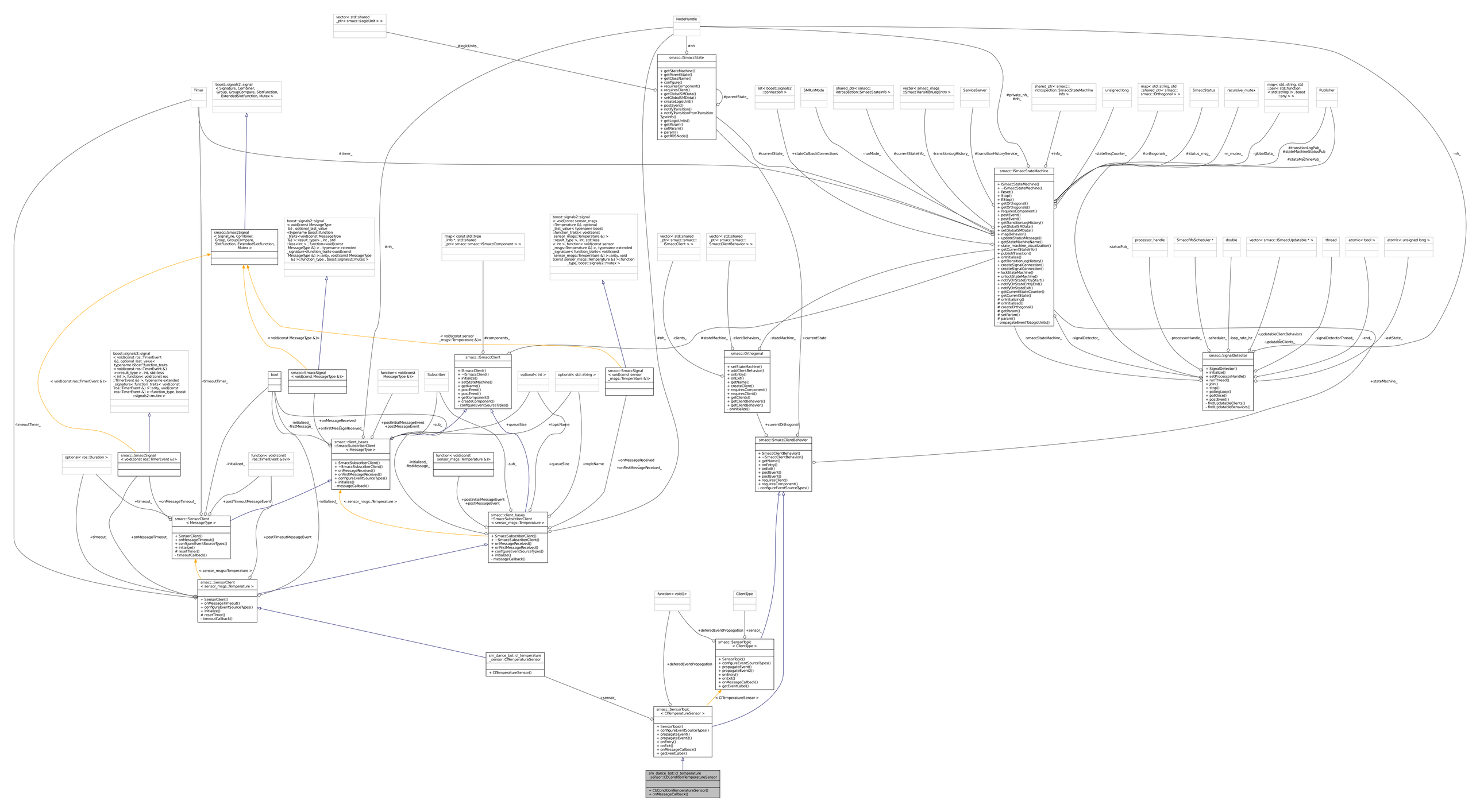
vector< std::shared _ptr< smacc::LogicUnit > >
NodeHandle
#logicUnits_
#nh
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createLogicUnit()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getLogicUnits()
+ getParam()
+ setParam()
+ params()
+ getROSNode()
#parentState_
Timer
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
list< boost::signals2 ::connection >
SMRunMode
shared_ptr< smacc:: introspection::SmaccStateInfo >
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
#private_nh_
#nh_
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
unsigned long
map< std::string, std ::shared_ptr< smacc:: smacc::Orthogonal > >
SmaccStatus
recursive_mutex
map< std::string, std ::bair< std::function < std::string()>, boost ::any> > >
Publisher
#timer_
#currentState_
+stateCallbackConnections
-runMode_
#currentStateInfo_
-transitionLogHistory_
#transitionHistoryService_
+info_
-stateSeqCounter_
#orthogonals_
#status_msg_
-m_mutex_
-globalData_
#transitionLogPub
#stateMachineStatusPub
#stateMachinePub_
-nh_
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()
+ createSignalConnection()
+ lockStateMachine()
+ unlockStateMachine()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnStateExit()
+ getCurrentStateCounter()
+ getCurrentState()
# onInitializing()
# oninitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- propagateEventToLogicUnits()
-statusPub_
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
boost::signals2::signal < void(const MessageType &) , optional_last_value <typename boost::function_traits <void(const MessageType &) >::result_type> , int , std ::less<int > , function<void(const MessageType &) > , typename extended _signature<function_traits<void(const MessageType &) >::arity, void(const MessageType &) >::function_type , boost::signals2::mutex >
map< const std::type _info *, std::shared _ptr< smacc::smacc::ISmaccComponent > >
boost::signals2::signal < void(const sensor _msgs::Temperature &), optional _last_value< typename boost ::function_traits< void(const sensor _msgs::Temperature &) > ::result_type >, int, std::less < int >, function< void(const sensor _msgs::Temperature &) >, typename extended _signature< function_traits< void(const sensor_msgs::Temperature &) >::arity, void (const sensor_msgs::Temperature &) >::function _type, boost::signals2::mutex >
vector< std::shared _ptr< smacc::smacc:: ISmaccClient > >
vector< std::shared _ptr< smacc::smacc:: SmaccClientBehavior > >
#components_
#nh_
-clients_
#stateMachine_
-clientBehaviors_
-stateMachine_
+currentState
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
+stateMachine_
-timeoutTimer_
< void(const MessageType &)>
< void(const sensor _msgs::Temperature &)>
boost::signals2::signal < void(const ros::TimerEvent &), optional_last_value< typename boost::function_traits < void(const ros::TimerEvent &) >::result_type >, int, std::less < int >, function< void(const ros ::TimerEvent &) >, typename extended _signature< function_traits< void(const ros::TimerEvent &) >::arity, void(const ros::TimerEvent &) >::function_type, boost ::signals2::mutex >
< void(const ros::TimerEvent &)>
bool
smacc::SmaccSignal < void(const MessageType &)>
function< void(const MessageType &)>
Subscriber
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ onEntry()
+ postEvent()
+ getComponent()
+ createComponent()
- configureEventSourceTypes()
optional< int >
optional< std::string >
smacc::SmaccSignal < void(const sensor _msgs::Temperature &)>
smacc::Orthogonal
+ setStateMachine()
+ addClientBehavior()
+ onEntry()
+ onExit()
+ getName()
+ create()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ getClientBehavior()
- onInitialize()
+currentOrthogonal
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getName()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
- configureEventSourceTypes()
-initialized_
-firstMessage_
+onMessageReceived
+onFirstMessageReceived_
+postInitialMessageEvent
+postMessageEvent
-sub_
+queueSize
+topicName
+onMessageReceived
+onFirstMessageReceived_
smacc::client_bases ::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
function< void(const sensor_msgs::Temperature &)>
optional< ros::Duration >
smacc::SmaccSignal < void(const ros::TimerEvent &)>
function< void(const ros::TimerEvent &ev)>
+timeout_
+onMessageTimeout_
+postTimeoutMessageEvent
smacc::SensorClient < MessageType >
+ SensorClient()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
< sensor_msgs::Temperature >
smacc::client_bases ::SmaccSubscriberClient < sensor_msgs::Temperature >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SensorClient < sensor_msgs::Temperature >
+ SensorClient()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
function< void()>
ClientType
+deferedEventPropagation
+sensor_
smacc::SensorTopic < ClientType >
+ SensorTopic()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_temperature _sensor::CTemperatureSensor
+ CTemperatureSensor()
+sensor_
< CTemperatureSensor >
smacc::SensorTopic < CTemperatureSensor >
+ SensorTopic()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_temperature _sensor::CbConditionBorn1temperatureSensor
+ CbConditionTemperatureSensor()
+ onMessageCallback()