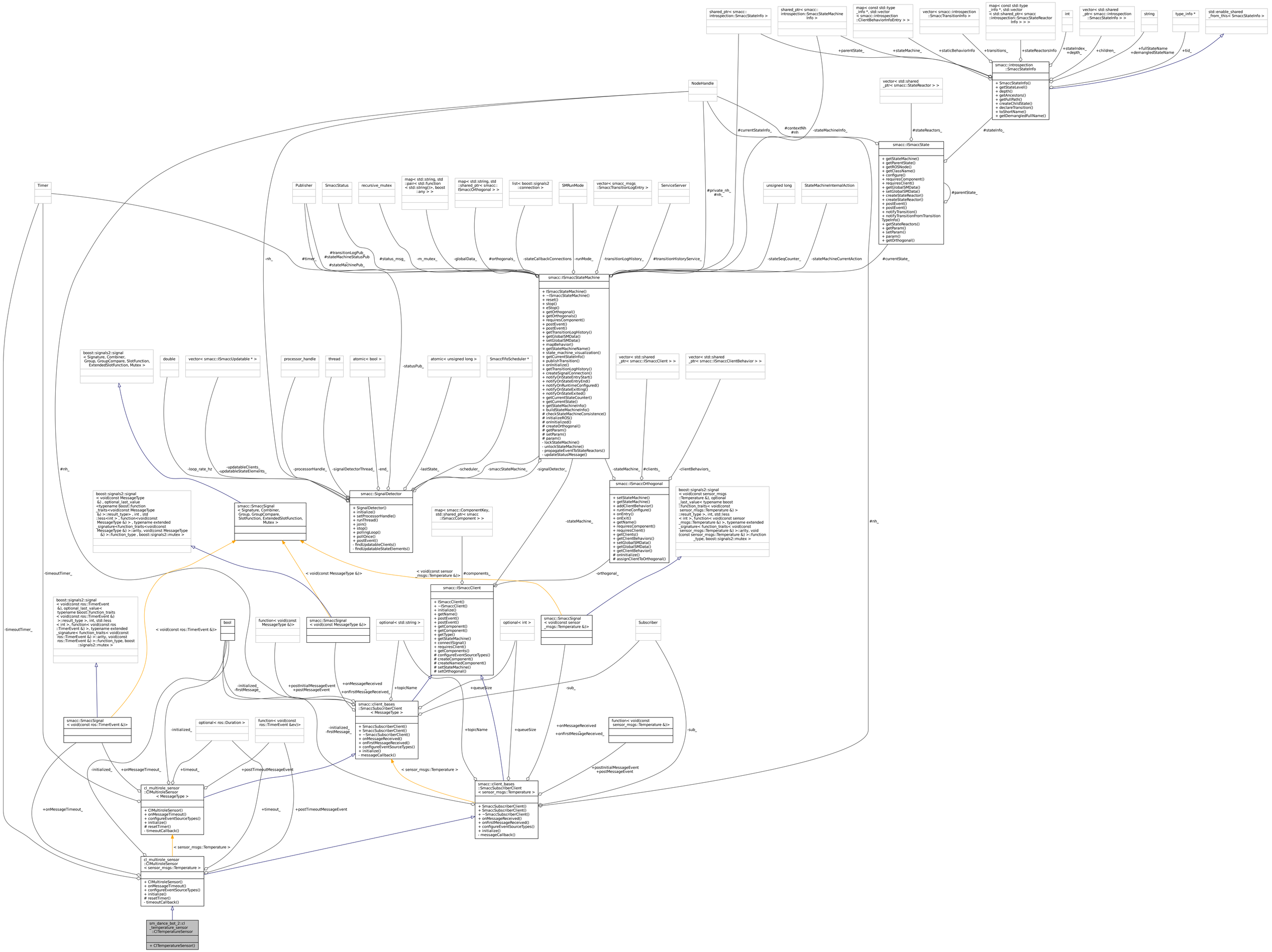
shared_ptr< smacc::introspection::SmaccStateInfo >
shared_ptr< smacc::introspection::SmaccStateMachineInfo >
map< const std::type_info *, std::vector< smacc::introspection::ClientBehaviorInfoEntry > >
vector< smacc::introspection::SmaccTransitionInfo >
map< const std::type_info *, std::vector< std::shared_ptr< smacc::introspection::SmaccStateReactorInfo > > >
int
vector< std::shared_ptr< smacc::introspection::SmaccStateInfo > >
string
type_info *
std::enable_shared_from_this< SmaccStateInfo >
+parentState_
+stateMachine_
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+stateIndex_
+depth_
+children_
+fullStateName
+demangledStateName
+tid_
smacc::introspection::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
NodeHandle
vector< std::shared_ptr< smacc::StateReactor > >
#currentStateInfo_
#contextNh #nh
-stateMachineInfo_
#stateReactors_
#stateInfo_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransitionTypeInfo()
+ getStateReactors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
#parentState_
Timer
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
StateMachineInternalAction
-nh_
#timer_
#transitionLogPub #stateMachineStatusPub #stateMachinePub_
#status_msg_
-m_mutex_
-globalData_
#orthogonals_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
-stateMachineCurrentAction
#currentState_
smacc::ISmaccStateMachine
+ 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()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
# checkStateMachineConsistence()
# initializeROS()
# oninitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
double
vector< smacc::ISmaccUpdatable * >
processor_handle
thread
atomic< bool >
-statusPub_
atomic< unsigned long >
SmaccFifoScheduler *
vector< std::shared_ptr< smacc::ISmaccClient > >
vector< std::shared_ptr< smacc::ISmaccClientBehavior > >
#nh_
-loop_rate_hz
-updatableClients_ -updatableStateElements_
-processorHandle_
-signalDetectorThread_
-end_
-lastState_
-scheduler_
-smaccStateMachine_
-signalDetector_
-stateMachine_
#clients_
-clientBehaviors_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
smacc::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
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 >
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
map< smacc::ComponentKey, std::shared_ptr< 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 >
-timeoutTimer_
< void(const MessageType &)>
< void(const sensor_msgs::Temperature &)>
#components_
-orthogonal_
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# configureEventSourceTypes()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
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 >
bool
function< void(const MessageType &)>
smacc::SmaccSignal < void(const MessageType &)>
optional< std::string >
optional< int >
smacc::SmaccSignal < void(const sensor_msgs::Temperature &)>
Subscriber
< void(const ros::TimerEvent &)>
-initialized_ -firstMessage_
+postInitialMessageEvent +postMessageEvent
+onMessageReceived
+onFirstMessageReceived_
+topicName
+queueSize
-sub_
smacc::client_bases::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SmaccSignal < void(const ros::TimerEvent &)>
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
-initialized_
-initialized_ +onMessageTimeout_
+timeout_
+postTimeoutMessageEvent
+onMessageReceived
+onFirstMessageReceived_
function< void(const sensor_msgs::Temperature &)>
-sub_
+topicName
+queueSize
< sensor_msgs::Temperature >
+postInitialMessageEvent +postMessageEvent
cl_multirole_sensor::ClMultiroleSensor < MessageType >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
smacc::client_bases::SmaccSubscriberClient < sensor_msgs::Temperature >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
+onMessageTimeout_
+timeout_
+postTimeoutMessageEvent
< sensor_msgs::Temperature >
cl_multirole_sensor::ClMultiroleSensor < sensor_msgs::Temperature >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
sm_dance_bot_2::cl_temperature_sensor::ClTemperatureSensor
+ ClTemperatureSensor()