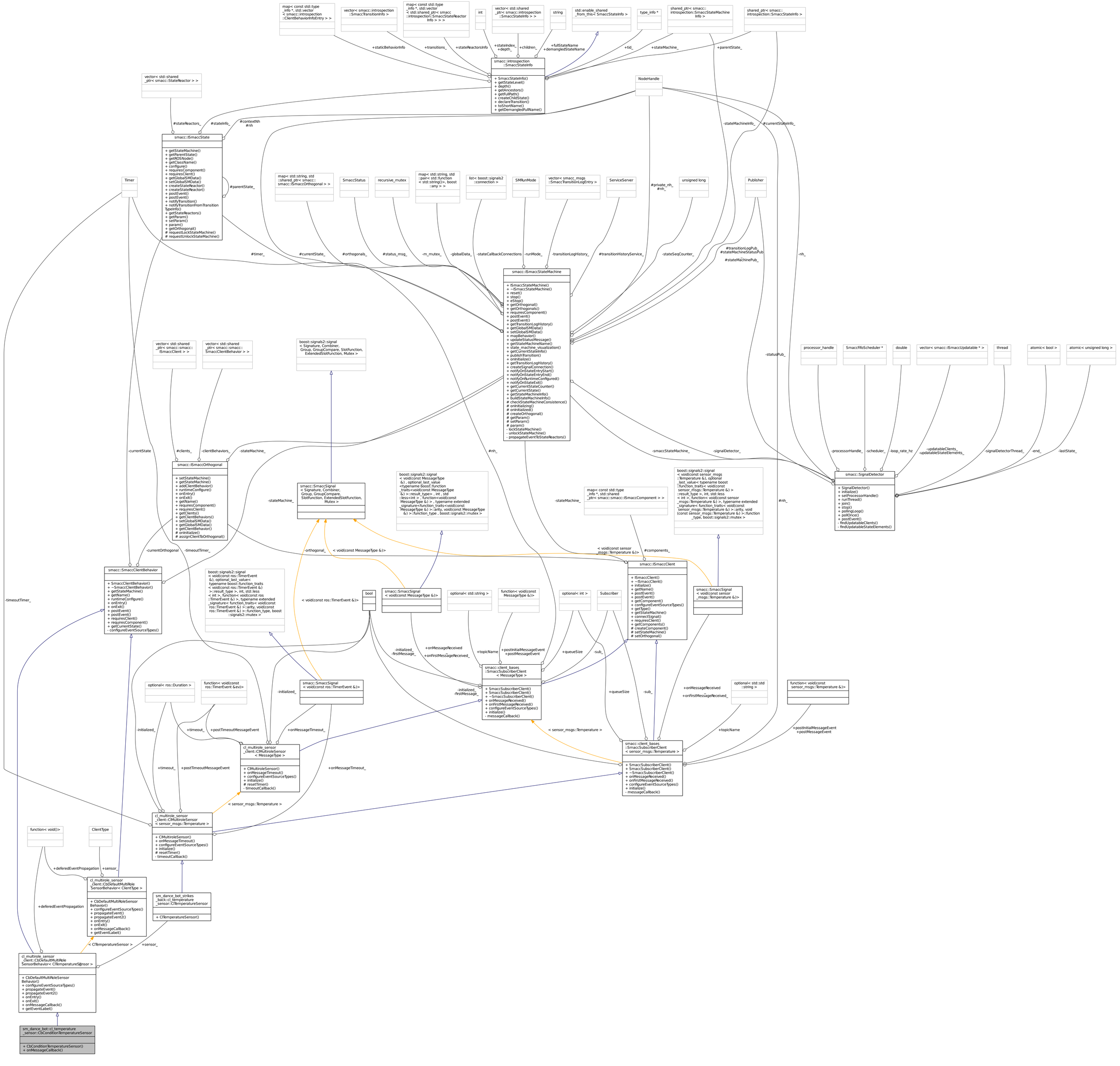
smacc::introspection::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition TypeInfo()
+ getStateReactors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
# requestLockStateMachine()
# requestUnlockStateMachine()
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()
smacc::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getStateMachine()
+ getName()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
+ getCurrentState()
- configureEventSourceTypes()
smacc::SmaccSignal
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# createComponent()
# setStateMachine()
# setOrthogonal()
smacc::client_bases ::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::client_bases ::SmaccSubscriberClient < sensor_msgs::Temperature >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
cl_multirole_sensor _client::CIMultiroleSensor < MessageType >
+ CIMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
cl_multirole_sensor _client::CIMultiroleSensor < sensor_msgs::Temperature >
+ CIMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
cl_multirole_sensor _client::CbDefaultMultiRole SensorBehavior< ClientType >
+ CbDefaultMultiRoleSensor Behavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot_strikes _back::cl_temperature _sensor::CITemperatureSensor
+ CITemperatureSensor()
cl_multirole_sensor _client::CbDefaultMultiRole SensorBehavior< CITemperatureSensor >
+ CbDefaultMultiRoleSensor Behavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_temperature _sensor::CbConditionTemperatureSensor
+ CbConditionTemperatureSensor()
+ onMessageCallback()
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
NodeHandle
Timer
Publisher
vector< std::shared _ptr< smacc::StateReactor > >
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
vector< std::shared _ptr< smacc::introspection ::SmaccStateInfo > >
string
std::enable_shared _from_this< SmaccStateInfo >
type_info *
shared_ptr< smacc:: introspection::SmaccStateMachine Info >
shared_ptr< smacc:: introspection::SmaccStateInfo >
map< std::string, std ::shared_ptr< smacc:: smacc::ISmaccOrthogonal > >
SmaccStatus
recursive_mutex
map< std::string, std ::pair< std::function < std::string()>, boost ::any > >
list< boost::signals2 ::connection >
SMRunMode
vector< smacc_msgs ::SmaccTransitionLogEntry >
ServiceServer
unsigned long
vector< std::shared _ptr< smacc::smacc:: ISmaccClient > >
vector< std::shared _ptr< smacc::smacc:: SmaccClientBehavior > >
boost::signals2::signal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
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 >
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
optional< std::string >
function< void(const MessageType &)>
optional< int >
Subscriber
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
function< void()>
ClientType
optional< std::std ::string >
function< void(const sensor_msgs::Temperature &)>
#stateReactors_
#stateInfo_
#contextNh #nh
#parentState_
#timer_
#currentState_
#orthogonals_
#status_msg_
-m_mutex_
-globalData_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
#transitionLogPub_
#stateMachineStatusPub_
#stateMachinePub_
-nh_
-statusPub_
#private_nh_ #nh_
-stateMachineInfo_
#currentStateInfo_
-currentState
#clients_
-clientBehaviors_
-stateMachine_
#nh_
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClients_ -updatableStateElements_
-signalDetectorThread_
-end_
-lastState_
-currentOrthogonal
-timeoutTimer_
-orthogonal_
< void(const MessageType &)>
< void(const sensor _msgs::Temperature &)>
#components_
< void(const ros::TimerEvent &)>
-initialized_ -firstMessage_
+onMessageReceived
+onFirstMessageReceived_
+topicName
+postInitialMessageEvent +postMessageEvent
+queueSize
-sub_
+timeout_
+postTimeoutMessageEvent
+onMessageTimeout_
< sensor_msgs::Temperature >
+deferedEventPropagation
+sensor_
< CITemperatureSensor >