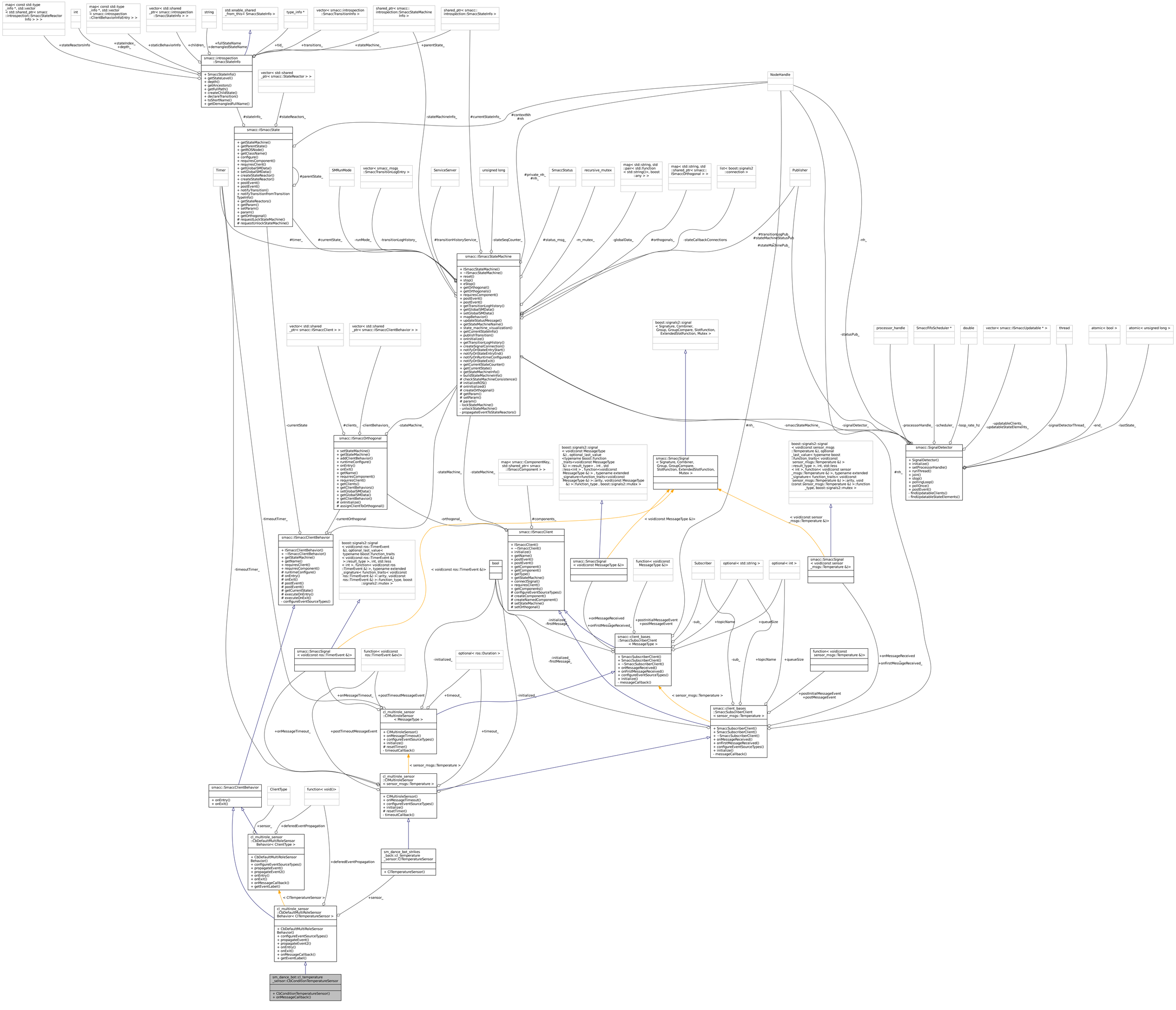
map< const std::type_info *, std::vector< std::shared_ptr< smacc::introspection::SmaccStateReactorInfo > > >
int
map< const std::type_info *, std::vector< smacc::introspection::ClientBehaviorInfoEntry > >
vector< std::shared_ptr< smacc::introspection::SmaccStateInfo > >
string
std::enable_shared_from_this< SmaccStateInfo >
type_info *
vector< smacc::introspection::SmaccTransitionInfo >
shared_ptr< smacc::introspection::SmaccStateMachineInfo >
shared_ptr< smacc::introspection::SmaccStateInfo >
+stateReactorsInfo
+stateIndex_
+depth_
+staticBehaviorInfo
+children_
+fullStateName
+demangledStateName
+tid_
+transitions_
+stateMachine_
+parentState_
smacc::introspection::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
vector< std::shared_ptr< smacc::StateReactor > >
#stateInfo_
#stateReactors_
NodeHandle
-stateMachineInfo_
#currentStateInfo_
#contextNh
#nh
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransitionTypeInfo()
+ getStateReactors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
# requestLockStateMachine()
# requestUnlockStateMachine()
#parentState_
Timer
SMRunMode
vector< smacc_msgs::SmaccTransitionLogEntry >
ServiceServer
unsigned long
#private_nh_
#nh_
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 >
Publisher
#timer_
#currentState_
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
#status_msg_
-m_mutex_
-globalData_
#orthogonals_
-stateCallbackConnections
#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()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExit()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
vector< std::shared_ptr< smacc::ISmaccClient > >
vector< std::shared_ptr< smacc::ISmaccClientBehavior > >
boost::signals2::signal< Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
-statusPub_
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
-currentState
#clients_
-clientBehaviors_
-stateMachine_
#nh_
-smaccStateMachine_
-signalDetector_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClients_
-updatableStateElements_
-signalDetectorThread_
-end_
-lastState_
smacc::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
smacc::SmaccSignal
< Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
map< smacc::ComponentKey, std::shared_ptr< smacc::ISmaccComponent > >
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 >
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 &) >::arity, void (const sensor_msgs::Temperature &) >::function_type, boost::signals2::mutex >
-timeoutTimer_
-currentOrthogonal
-orthogonal_
#components_
< void(const MessageType &)>
< void(const sensor_msgs::Temperature &)>
smacc::ISmaccClientBehavior
+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# getCurrentState()
# executeOnEntry()
# executeOnExit()
- configureEventSourceTypes()
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 &) >::arity, void(const ros::TimerEvent &) >::function_type, boost::signals2::mutex >
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# configureEventSourceTypes()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
bool
< void(const ros::TimerEvent &)>
smacc::SmaccSignal
< void(const MessageType &)>
function< void(const MessageType &)>
Subscriber
optional< std::string >
optional< int >
smacc::SmaccSignal
< void(const sensor_msgs::Temperature &)>
smacc::SmaccSignal
< void(const ros::TimerEvent &)>
function< void(const ros::TimerEvent &ev)>
optional< ros::Duration >
-initialized_
-firstMessage_
+onMessageReceived
+onFirstMessageReceived_
+postInitialMessageEvent
+postMessageEvent
-sub_
+topicName
+queueSize
smacc::client_bases
::SmaccSubscriberClient
< MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
function< void(const sensor_msgs::Temperature &)>
+onMessageTimeout_
+postTimeoutMessageEvent
+timeout_
cl_multirole_sensor
::ClMultiroleSensor
< MessageType >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
< sensor_msgs::Temperature >
smacc::client_bases
::SmaccSubscriberClient
< sensor_msgs::Temperature >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
cl_multirole_sensor
::ClMultiroleSensor
< sensor_msgs::Temperature >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
smacc::SmaccClientBehavior
+ onEntry()
+ onExit()
ClientType
function< void()>
+sensor_
+deferedEventPropagation
cl_multirole_sensor
::CbDefaultMultiRoleSensor
Behavior< ClientType >
+ CbDefaultMultiRoleSensor Behavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot_strikes
_back::cl_temperature
_sensor::ClTemperatureSensor
+ ClTemperatureSensor()
< ClTemperatureSensor >
+sensor_
cl_multirole_sensor
::CbDefaultMultiRoleSensor
Behavior< ClTemperatureSensor >
+ CbDefaultMultiRoleSensor Behavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_temperature
_sensor::CbConditionTemperatureSensor
+ CbConditionTemperatureSensor()
+ onMessageCallback()