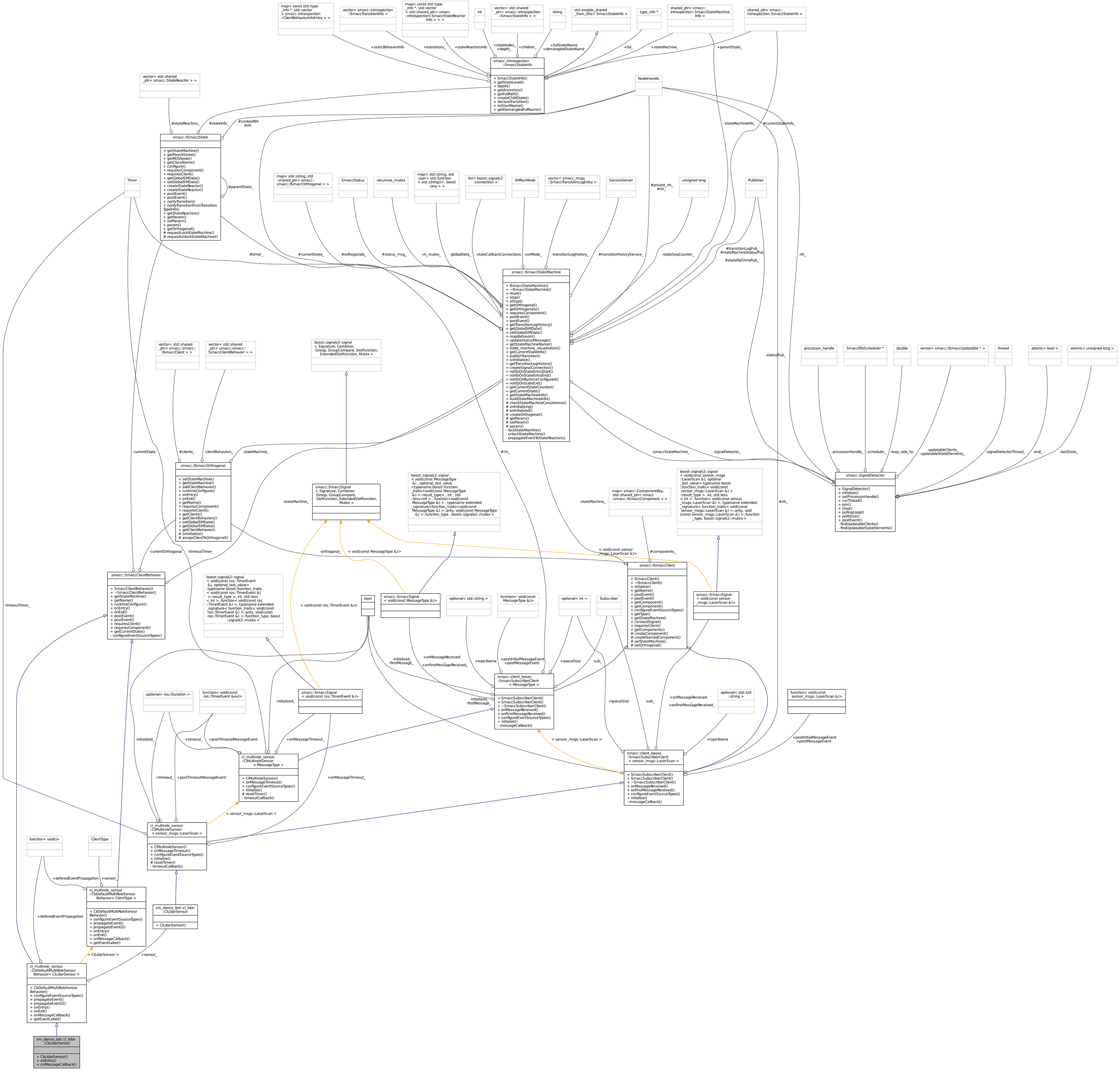
smacc::introspection::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransitionTypeInfo()
+ 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::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
smacc::SmaccClientBehavior
+ SmaccClientBehavior()
+ ~SmaccClientBehavior()
+ getStateMachine()
+ getName()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ postEvent()
+ postEvent()
+ requiresClient()
+ requiresComponent()
+ getCurrentState()
- configureEventSourceTypes()
smacc::client_bases::SmaccSubscriberClient < MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::client_bases::SmaccSubscriberClient < sensor_msgs::LaserScan >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
cl_multirole_sensor::ClMultiroleSensor < MessageType >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
cl_multirole_sensor::ClMultiroleSensor < sensor_msgs::LaserScan >
+ ClMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
cl_multirole_sensor::CbDefaultMultiRoleSensorBehavior< ClientType >
+ CbDefaultMultiRoleSensorBehavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
cl_multirole_sensor::CbDefaultMultiRoleSensorBehavior< ClLidarSensor >
+ CbDefaultMultiRoleSensorBehavior()
+ configureEventSourceTypes()
+ propagateEvent()
+ propagateEvent2()
+ onEntry()
+ onExit()
+ onMessageCallback()
+ getEventLabel()
sm_dance_bot::cl_lidar::ClLidarSensor
+ ClLidarSensor()
sm_dance_bot::cl_lidar::CbLidarSensor
+ CbLidarSensor()
+ onEntry()
+ onMessageCallback()
smacc::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
smacc::SmaccSignal < void(const MessageType &)>
smacc::SmaccSignal < void(const ros::TimerEvent &)>
smacc::SmaccSignal < void(const sensor_msgs::LaserScan &)>
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
std::enable_shared_from_this< SmaccStateInfo >
type_info *
shared_ptr< smacc::introspection::SmaccStateMachineInfo >
shared_ptr< smacc::introspection::SmaccStateInfo >
vector< std::shared_ptr< smacc::StateReactor > >
NodeHandle
Timer
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
Publisher
vector< std::shared_ptr< 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< smacc::ComponentKey, std::shared_ptr< smacc::smacc::ISmaccComponent > >
boost::signals2::signal < void(const sensor_msgs::LaserScan &), optional_last_value< typename boost::function_traits< void(const sensor_msgs::LaserScan &) >::result_type >, int, std::less < int >, function< void(const sensor_msgs::LaserScan &) >, typename extended_signature< function_traits< void(const sensor_msgs::LaserScan &) >::arity, void (const sensor_msgs::LaserScan &) >::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< std::string >
function< void(const sensor_msgs::LaserScan &)>
optional< ros::Duration >
function< void(const ros::TimerEvent &ev)>
function< void()>
ClientType
#stateReactors_
#stateInfo_
#contextNh #nh
#parentState_
#timer_
#currentState_
#orthogonals_
#status_msg_
-m_mutex_
-globalData_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-stateSeqCounter_
#transitionLogPub #stateMachineStatusPub_
#stateMachinePub_
-nh_
-statusPub_
-stateMachineInfo_
#currentStateInfo_
#private_nh_ #nh_
-currentState
#clients_
-clientBehaviors_
-stateMachine_
#nh_
-stateMachine_
-orthogonal_
< void(const MessageType &)>
-currentOrthogonal
-timeoutTimer_
-timeoutTimer_
< void(const ros::TimerEvent &)>
-initialized_
-firstMessage_
+onMessageReceived
+onFirstMessageReceived_
+topicName
+postInitialMessageEvent +postMessageEvent
+queueSize
-sub_
-initialized_
+timeout_
+postTimeoutMessageEvent
+onMessageTimeout_
< sensor_msgs::LaserScan >
+deferedEventPropagation
+sensor_
< ClLidarSensor >
-processorHandle_
-scheduler_
-loop_rate_hz
-updatable_clients_ -updatable_state_elements_
-signal_detector_thread_
-end_
-last_state_
-saccStateMachine_
-signalDetector_
-stateMachine_
#components_
< void(const sensor_msgs::LaserScan &)>
+postInitialMessageEvent +postMessageEvent