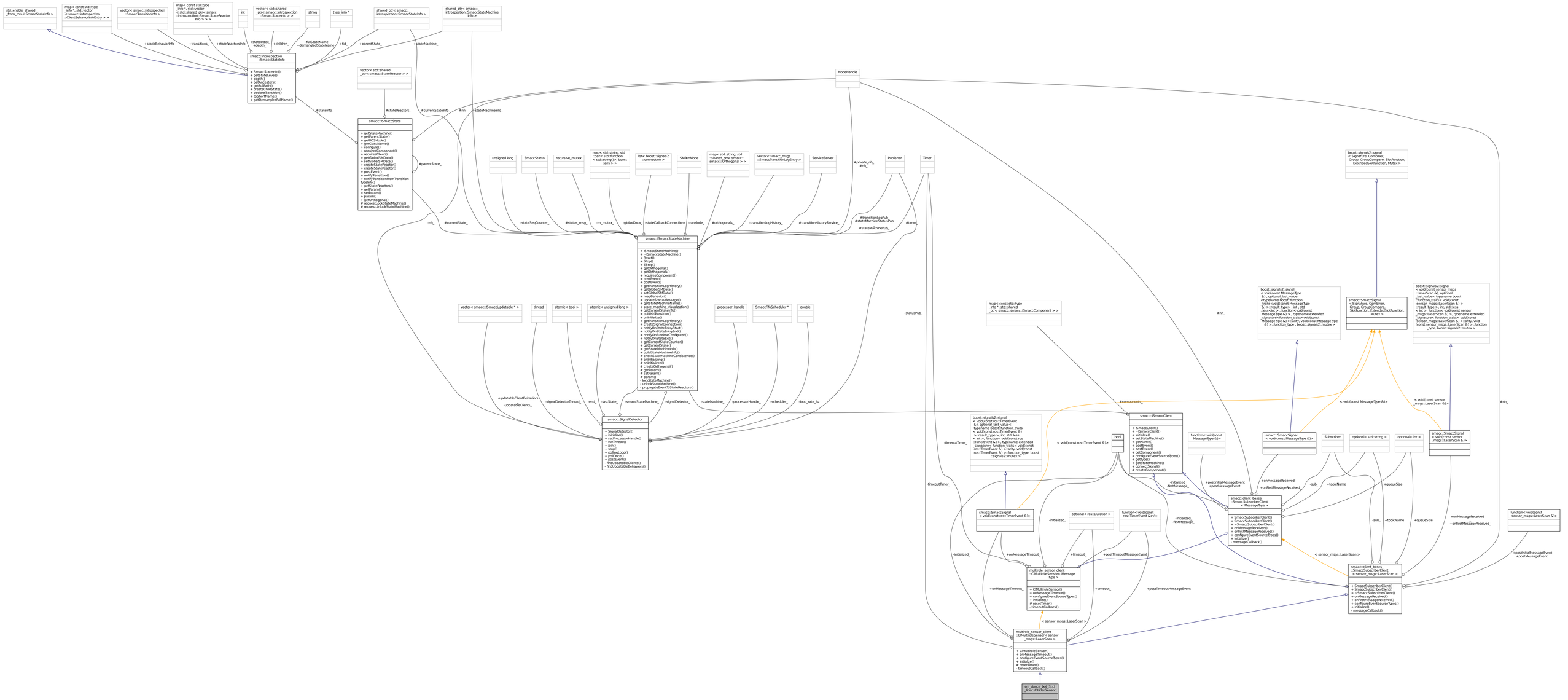
std::enable_shared_from_this< SmaccStateInfo >
map< const std::type_info *, std::vector< smacc::introspection::ClientBehaviorInfoEntry > >
vector< smacc::introspection::SmaccTransitionInfo >
map< const std::type_info *, std::vector< std::shared_ptr< smacc::introspection::SmaccStateReactorInfo > > >
int
vector< std::shared_ptr< smacc::introspection::SmaccStateInfo > >
string
type_info *
shared_ptr< smacc::introspection::SmaccStateInfo >
shared_ptr< smacc::introspection::SmaccStateMachineInfo >
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+stateIndex_
+depth_
+children_
+fullStateName
+demangledStateName
+tid_
+parentState_
+stateMachine_
smacc::introspection::SmaccStateInfo
+ SmaccStateInfo()
+ getState(Level)
+ depth()
+ getAncestors()
+ getFullPath()
+ checkChildState()
+ getReactorInfos()
+ toShortName()
+ getDemangledFullName()
vector< std::shared_ptr< smacc::IStateReactor > >
NodeHandle
#stateInfo_
#stateReactors_
#currentStateInfo_
#nh_
-stateMachineInfo_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getStateName()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createSiStateReactor()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition
+ getStates()
+ getStateReactors()
+ getParam()
+ setParam()
+ parent()
+ requestLock()
# requestLockStateMachine()
# requestUnlockStateMachine()
#parentState_
unsigned long
SmaccStatus
recursive_mutex
map< std::string, std::pair< std::function< std::string()>, boost::any > >
list< boost::signals2::connection >
SMRunMode
map< std::string, std::shared_ptr< smacc::IOrthogonal > >
vector< smacc_msgs::SmaccTransitionLogEntry >
ServiceServer
#private_nh_
#nh_
Publisher
Timer
boost::signals2::signal< Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex >
-nh_
#currentState_
-stateSeqCounter_
#status_msg_
-m_mutex_
-globalData_
-stateCallbackConnections
-runMode_
#orthogonals_
-transitionLogHistory_
#transitionHistoryService_
#transitionLogPub_
#stateMachineStatusPub
#stateMachinePub_
#timer_
smacc::ISmaccStateMachine
+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ Reset()
+ Stop()
+ Kill()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ publishStateMachineInfo()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnections()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ getStateMachine()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
# checkStateMachineConsistence()
# onInitializing()
# onInitialize()
# createOrthogonal()
# getParam()
# setParam()
# getStates()
# lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
-statusPub_
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
processor_handle
SmaccFRiVScheduler *
double
map< const std::type_info *, std::shared_ptr< smacc::ISmaccComponent > >
#components_
-updatableClientBehaviors
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
-smaccStateMachine_
-signalDetector_
-stateMachine_
-processorHandle_
-scheduler_
-loop_rate_hz
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ run(Thread)
+ stop()
+ join()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getClient()
+ postEvent()
+ getEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
+ getStateMachine()
+ getSmaccSignal()
# createComponent()
-timeoutTimer_
boost::signals2::signal< void(const ros::TimerEvent &)>
boost::signals2::signal< void(const MessageType &), boost::function_traits< void(const MessageType &)> >
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 MessageType &) > , typename extended_signature< function_traits< void(const MessageType &)> ::arity , void(const MessageType &) > ::function_type , boost::signals2::mutex >
< void(const ros::TimerEvent &)>
< void(const MessageType &)>
< void(const sensor_msgs::LaserScan &)>
#nh_
smacc::SmaccSignal< void(const ros::TimerEvent &)>
smacc::SmaccSignal< void(const MessageType &)>
smacc::SmaccSignal< void(const sensor_msgs::LaserScan &)>
bool
function< void(const MessageType &)>
Subscriber
optional< std::string >
optional< int >
-initialized_
-firstMessage_
+postInitialMessageEvent
+postMessageEvent
+onMessageReceived
+onFirstMessageReceived_
-sub_
+topicName
+queueSize
smacc::client_bases::SmaccSubscriberClient< MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
optional< ros::Duration >
function< void(const ros::TimerEvent &evt)>
-initialized_
-firstMessage_
+onMessageTimeout_
+timeout_
+postTimeoutMessageEvent
multirole_sensor_client::CiMultiroleSensor< MessageType >
+ CiMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
< sensor_msgs::LaserScan >
smacc::client_bases::SmaccSubscriberClient< sensor_msgs::LaserScan >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
multirole_sensor_client::CiMultiroleSensor< sensor_msgs::LaserScan >
+ CiMultiroleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
sm_dorsal_bot_3::sl_ldar::CiLidarSensor
+ CiLidarSensor()