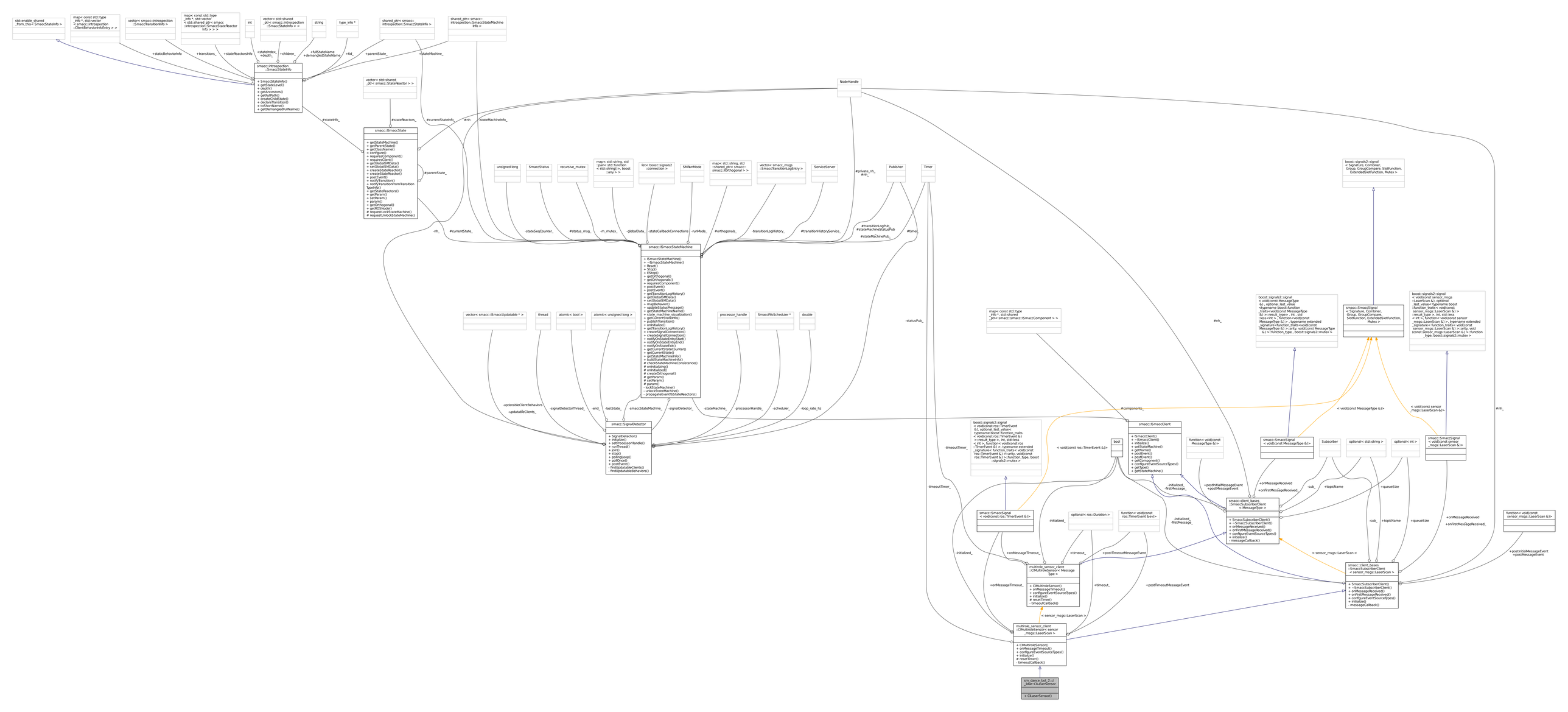
std::enable_shared
_from_this< SmaccStateInfo >
map< const std::type
_info *, std::vector
< smacc::introspection
::ClientBehaviorInfoEntry > >
vector< smacc::introspection
::SmaccTransitionInfo >
map< const std::type
_info *, std::vector
< std::shared_ptr< smacc
::introspection::SmaccStateReactor
Info > > >
int
vector< std::shared
_ptr< smacc::introspection
::SmaccStateInfo > >
string
type_info *
shared_ptr< smacc::
introspection::SmaccStateInfo >
shared_ptr< smacc::
introspection::SmaccStateMachine
Info >
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+stateIndex,
+depth,
+children,
+fullStateName
+demangledStateName
+tid,
+parentState_
+stateMachine_
smacc::introspection
::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ getNextChildState()
+ getNextChildIdx()
+ toShortName()
+ getDemangledFullName()
vector< std::shared
_ptr< smacc::StateReactor > >
NodeHandle
#stateInfo_
#stateReactors_
#currentStateInfo_
#nh
-stateMachineInfo_
smacc::ISmaccState
+ getStateMachine()
+ getParentState()
+ getStateReactor()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ postEvent()
+ notify(Transition)
+ notify(TransitionFrom
Type,...)
+ getStateReactors()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getOrthogonals()
# 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::
smacc::IOrthogonal > >
vector< smacc::msgs
::SmaccTransitionLogEntry >
ServiceServer
Publisher
Timer
#private_nh_
#nh_
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()
+ getGlobalSMData()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ updateStatusMessage()
+ publishStateMachineInfo()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ createSignalConnection()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ getStateMachine()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
# checkStateMachineConsistence()
# onInitializing()
# onInitialize()
# createOrthogonal()
# getParam()
# setParam()
# param()
# ~ISmaccStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
processor_handle
SmaccFifoScheduler *
double
-updatableClientBehaviors_
-updatableClients_
-signalDetectorThread_
-end_
-lastState_
-smaccStateMachine_
-signalDetector_
-stateMachine_
-processorHandle_
-scheduler_
-loop_rate_hz
smacc::SignalDetector
+ SignalDetector()
+ initMaster()
+ getProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ postEvent()
- findUpdatableClients()
- findUpdatableBehaviors()
-statusPub_
map< const std::type
_info *, std::shared
_ptr< smacc::ISmaccComponent > >
#components_
boost::signals2::signal
< void(const ros::TimerEvent
&), optional_last_value<
typename boost::function_traits
< void(const ros::TimerEvent &)
> ::result_type >, std::less
< int >, std::allocator
< std::pair< int, boost::function
< void(const ros::TimerEvent &) >>, typename extended
_signature< function_traits< void(const
ros::TimerEvent &) >::arity, void(const
ros::TimerEvent &) >::function_type, boost
::signals2::mutex >
-timeoutTimer_
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ setStateMachine()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
+ getStateMachine()
bool
< void(const ros::TimerEvent &)>
function< void(const
Message&type &)>
smacc::SmaccSignal
< void(const Message&type &)>
Subscriber
optional< std::string >
optional< int >
smacc::SmaccSignal
< void(const sensor
_msgs::LaserScan &)>
< void(const Message&type &)>
< void(const sensor
_msgs::LaserScan &)>
-initialized_
-firstMessage_
+postInitialMessageEvent
+postMessageEvent
+onMessageReceived
+onFirstMessageReceived_
-sub_
+topicName
+queueSize
smacc::client_bases
::SmaccSubscriberClient
< MessageType >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
smacc::SmaccSignal
< void(const ros::TimerEvent &)>
optional< ros::Duration >
function< void(const
ros::TimerEvent &evt)>
function< void(const
sensor_msgs::LaserScan &)>
-initialized_
+onMessageTimeout_
+timeout_
+postTimeoutMessageEvent
multirole_sensor_client
::CMultiRoleSensor< Message
Type >
+ CMultiRoleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
< sensor_msgs::LaserScan >
smacc::client_bases
::SmaccSubscriberClient
< sensor_msgs::LaserScan >
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
multirole_sensor_client
::CMultiRoleSensor< sensor
_msgs::LaserScan >
+ CMultiRoleSensor()
+ onMessageTimeout()
+ configureEventSourceTypes()
+ initialize()
# resetTimer()
- timeoutCallback()
sim_dance_bot_2::cl
_lidar::CLaserSensor
+ CLaserSensor()