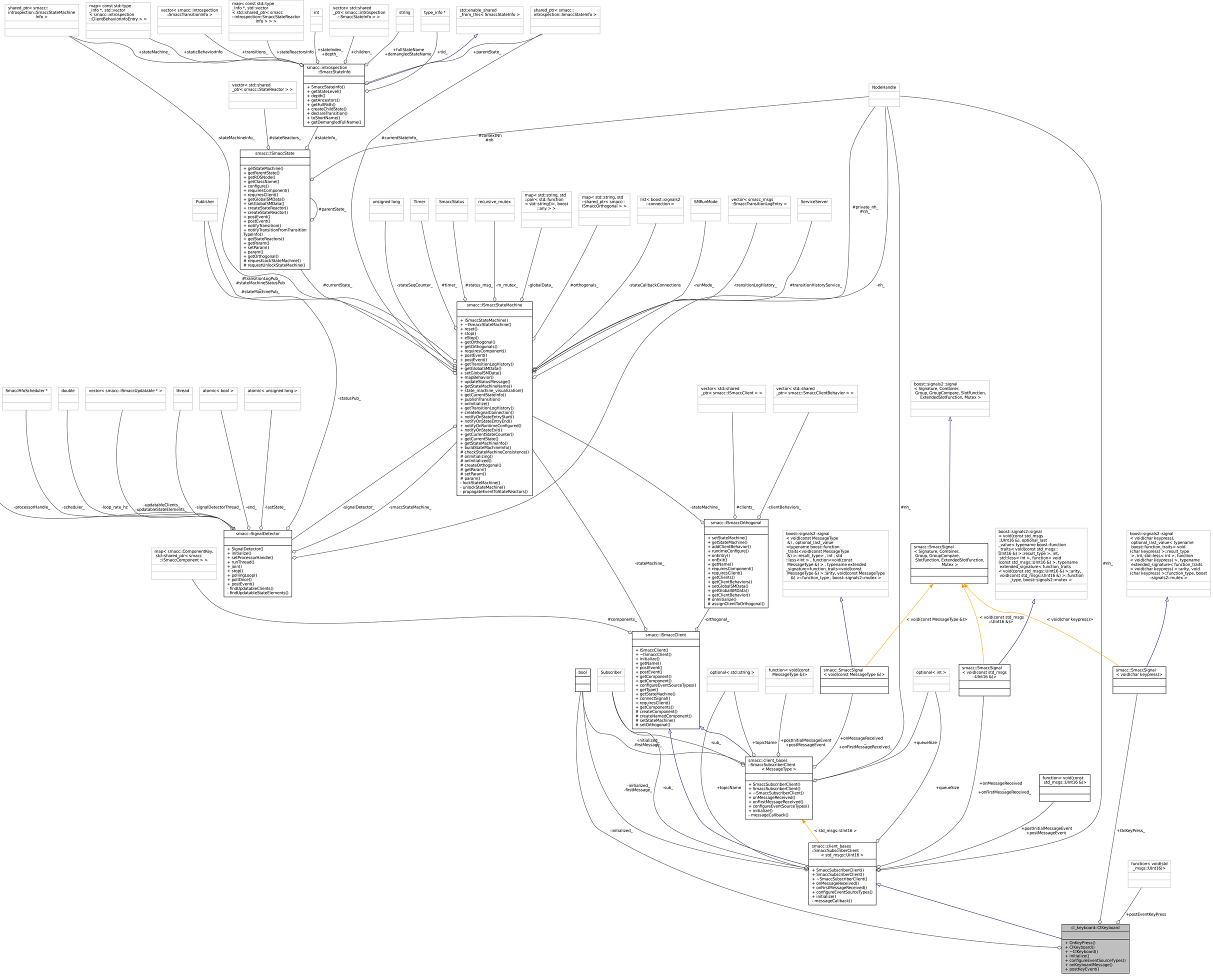
shared_ptr< smacc::
introspection::SmaccStateMachine
Info >
map< const std::type
_info *, std::vector
< smacc::introspection
::ClientBehaviorInfoEntry > >
vector< smacc::introspection
::SmaccTransitionInfo >
map< const std::type
_info *, std::vector
< std::shared_ptr< smacc
::introspection::SmaccStateReactor
Info > > >
int
vector< std::shared
_ptr< smacc::introspection
::SmaccStateInfo > >
string
type_info *
std::enable_shared
_from_this< SmaccStateInfo >
shared_ptr< smacc::
introspection::SmaccStateInfo >
+stateMachine_
+staticBehaviorInfo
+transitions_
+stateReactorsInfo
+stateIndex_
+depth_
+children_
+fullStateName
+demangledStateName
+tid_
+parentState_
smacc::introspection
::SmaccStateInfo
+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
vector< std::shared
_ptr< smacc::StateReactor > >
-stateMachineInfo_
#stateReactors_
#stateInfo_
#currentStateInfo_
NodeHandle
#contextNh
#nh
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()
#parentState_
Publisher
unsigned long
Timer
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 >
SMRunMode
vector< smacc_msgs
::SmaccTransitionLogEntry >
ServiceServer
#private_nh_
#nh_
#transitionLogPub_
#stateMachineStatusPub
#stateMachinePub_
#currentState_
-stateSeqCounter_
#timer_
#status_msg_
-m_mutex_
-globalData_
#orthogonals_
-stateCallbackConnections
-runMode_
-transitionLogHistory_
#transitionHistoryService_
-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()
# onInitializing()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
processor_handle
SmaccFifoScheduler *
double
vector< smacc::ISmaccUpdatable * >
thread
atomic< bool >
atomic< unsigned long >
-statusPub_
-processorHandle_
-scheduler_
-loop_rate_hz
-updatableClients_
-updatableStateElements_
-signalDetectorThread_
-end_
-lastState_
-signalDetector_
-smaccStateMachine_
smacc::SignalDetector
+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
map< smacc::ComponentKey,
std::shared_ptr< smacc
::ISmaccComponent > >
vector< std::shared
_ptr< smacc::ISmaccClient > >
vector< std::shared
_ptr< smacc::SmaccClientBehavior > >
-stateMachine_
#clients_
-clientBehaviors_
smacc::ISmaccOrthogonal
+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
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
< Signature, Combiner,
Group, GroupCompare,
SlotFunction, ExtendedSlotFunction,
Mutex >
boost::signals2::signal
< void(const std_msgs
::UInt16 &), optional_last
_value< typename boost::function
_traits< void(const std_msgs::
UInt16 &) >::result_type >, int,
std::less< int >, function< void
(const std_msgs::UInt16 &) >, typename
extended_signature< function_traits
< void(const std_msgs::UInt16 &) >::arity,
void(const std_msgs::UInt16 &) >::function
_type, boost::signals2::mutex >
boost::signals2::signal
< void(char keypress),
optional_last_value< typename
boost::function_traits< void
(char keypress) >::result_type
>, int, std::less< int >, function
< void(char keypress) >, typename
extended_signature< function_traits
< void(char keypress) >::arity, void
(char keypress) >::function_type, boost
::signals2::mutex >
#components_
-orthogonal_
smacc::ISmaccClient
+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ configureEventSourceTypes()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
bool
Subscriber
optional< std::string >
function< void(const
MessageType &)>
smacc::SmaccSignal
< void(const MessageType &)>
optional< int >
smacc::SmaccSignal
< void(const std_msgs
::UInt16 &)>
smacc::SmaccSignal
< void(char keypress)>
< void(const MessageType &)>
< void(const std_msgs
::UInt16 &)>
< void(char keypress)>
-initialized_
-firstMessage_
-sub_
+topicName
+postInitialMessageEvent
+postMessageEvent
+onMessageReceived
+onFirstMessageReceived_
+queueSize
smacc::client_bases
::SmaccSubscriberClient
< MessageType >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
function< void(const
std_msgs::UInt16 &)>
< std_msgs::UInt16 >
smacc::client_bases
::SmaccSubscriberClient
< std_msgs::UInt16 >
+ SmaccSubscriberClient()
+ SmaccSubscriberClient()
+ ~SmaccSubscriberClient()
+ onMessageReceived()
+ onFirstMessageReceived()
+ configureEventSourceTypes()
+ initialize()
- messageCallback()
function< void(std
_msgs::UInt16)>
+OnKeyPress_
+postEventKeyPress
cl_keyboard::CIKeyboard
+ OnKeyPress()
+ CIKeyboard()
+ ~CIKeyboard()
+ initialize()
+ configureEventSourceTypes()
+ onKeyboardMessage()
+ postKeyEvent()