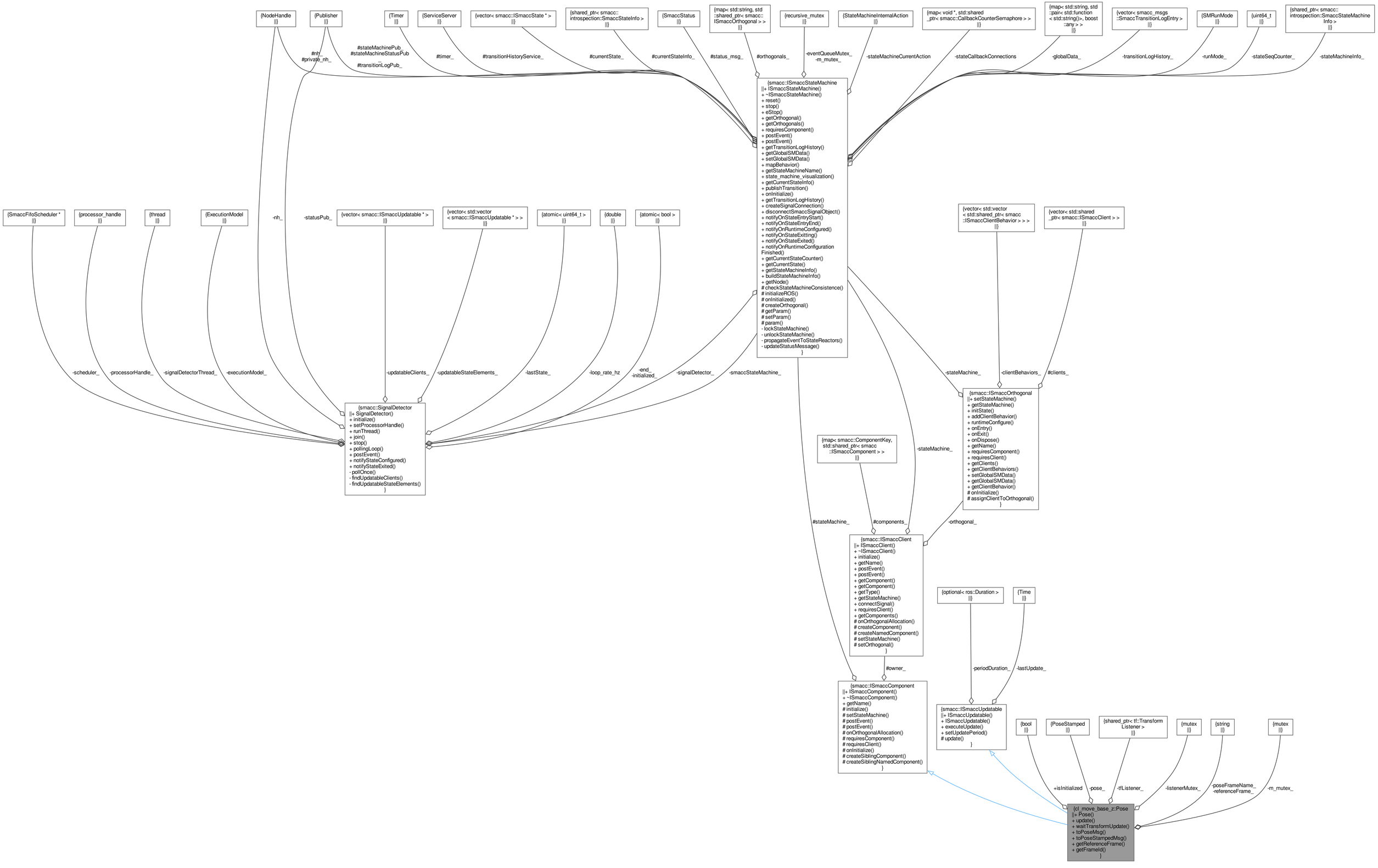
{NodeHandle ||}
{Publisher ||}
{Timer ||}
{ServiceServer ||}
{vector< smacc::ISmaccState * > ||}
{shared_ptr< smacc:: introspection::SmaccStateInfo > ||}
{SmaccStatus ||}
{map< std::string, std ::shared_ptr< smacc:: ISmaccOrthogonal > > ||}
{recursive_mutex ||}
{StateMachineInternalAction ||}
{map< void *, std::shared _ptr< smacc::CallbackCounterSemaphore > > ||}
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > ||}
{vector< smacc_msgs ::SmaccTransitionLogEntry > ||}
{SMRunMode ||}
{uint64_t ||}
{shared_ptr< smacc:: introspection::SmaccStateMachine Info > ||}
#nh_
#private_nh_
#stateMachinePub_
#stateMachineStatusPub _
#transitionLogPub_
#timer_
#transitionHistoryService_
#currentState_
#currentStateInfo_
#status_msg_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
-stateMachineInfo_
{smacc::ISmaccStateMachine ||+ ISmaccStateMachine() + ~ISmaccStateMachine() + reset() + stop() + eStop() + getOrthogonal() + getOrthogonals() + requiresComponent() + postEvent() + postEvent() + getTransitionLogHistory() + getGlobalSMData() + setGlobalSMData() + mapBehavior() + getStateMachineName() + state_machine_visualization() + getCurrentStateInfo() + publishTransition() + onInitialize() + getTransitionLogHistory() + createSignalConnection() + disconnectSmaccSignalObject() + notifyOnStateEntryStart() + notifyOnStateEntryEnd() + notifyOnRuntimeConfigured() + notifyOnStateExitting() + notifyOnStateExited() + notifyOnRuntimeConfiguration Finished() + getCurrentStateCounter() + getCurrentState() + getStateMachineInfo() + buildStateMachineInfo() + getNode() # checkStateMachineConsistence() # initializeROS() # onInitialized() # createOrthogonal() # getParam() # setParam() # param() - lockStateMachine() - unlockStateMachine() - propagateEventToStateReactors() - updateStatusMessage() }
{SmaccFifoScheduler * ||}
{processor_handle ||}
{thread ||}
{ExecutionModel ||}
-nh_
-statusPub_
{vector< smacc::ISmaccUpdatable * > ||}
{vector< std::vector < smacc::ISmaccUpdatable * > > ||}
{atomic< uint64_t > ||}
{double ||}
{atomic< bool > ||}
{vector< std::vector < std::shared_ptr< smacc ::ISmaccClientBehavior > > > ||}
{vector< std::shared _ptr< smacc::ISmaccClient > > ||}
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
-loop_rate_hz
-end_ -initialized_
-signalDetector_
-smaccStateMachine_
-stateMachine_
-clientBehaviors_
#clients_
{smacc::SignalDetector ||+ SignalDetector() + initialize() + setProcessorHandle() + runThread() + join() + stop() + pollingLoop() + postEvent() + notifyStateConfigured() + notifyStateExited() - pollOnce() - findUpdatableClients() - findUpdatableStateElements() }
{smacc::ISmaccOrthogonal ||+ setStateMachine() + getStateMachine() + initState() + addClientBehavior() + runtimeConfigure() + onEntry() + onExit() + onDispose() + getName() + requiresComponent() + requiresClient() + getClients() + getClientBehaviors() + setGlobalSMData() + getGlobalSMData() + getClientBehavior() # onInitialize() # assignClientToOrthogonal() }
{map< smacc::ComponentKey, std::shared_ptr< smacc ::ISmaccComponent > > ||}
-stateMachine_
#stateMachine_
#components_
-orthogonal_
{smacc::ISmaccClient ||+ ISmaccClient() + ~ISmaccClient() + initialize() + getName() + postEvent() + postEvent() + getComponent() + getComponent() + getType() + getStateMachine() + connectSignal() + requiresClient() + getComponents() # onOrthogonalAllocation() # createComponent() # createNamedComponent() # setStateMachine() # setOrthogonal() }
{optional< ros::Duration > ||}
{Time ||}
#owner_
-periodDuration_
-lastUpdate_
{smacc::ISmaccComponent ||+ ISmaccComponent() + ~ISmaccComponent() + getName() # initialize() # setStateMachine() # postEvent() # postEvent() # onOrthogonalAllocation() # requiresComponent() # requiresClient() # onInitialize() # createSiblingComponent() # createSiblingNamedComponent() }
{smacc::ISmaccUpdatable ||+ ISmaccUpdatable() + ISmaccUpdatable() + executeUpdate() + setUpdatePeriod() # update() }
{bool ||}
{PoseStamped ||}
{shared_ptr< tf::Transform Listener > ||}
{mutex ||}
{string ||}
{mutex ||}
+isInitialized
-pose_
-tfListener_
-listenerMutex_
-poseFrameName_ -referenceFrame_
-m_mutex_
{cl_move_base_z::Pose ||+ Pose() + update() + waitTransformUpdate() + toPoseMsg() + toPoseStampedMsg() + getReferenceFrame() + getFrameId() }