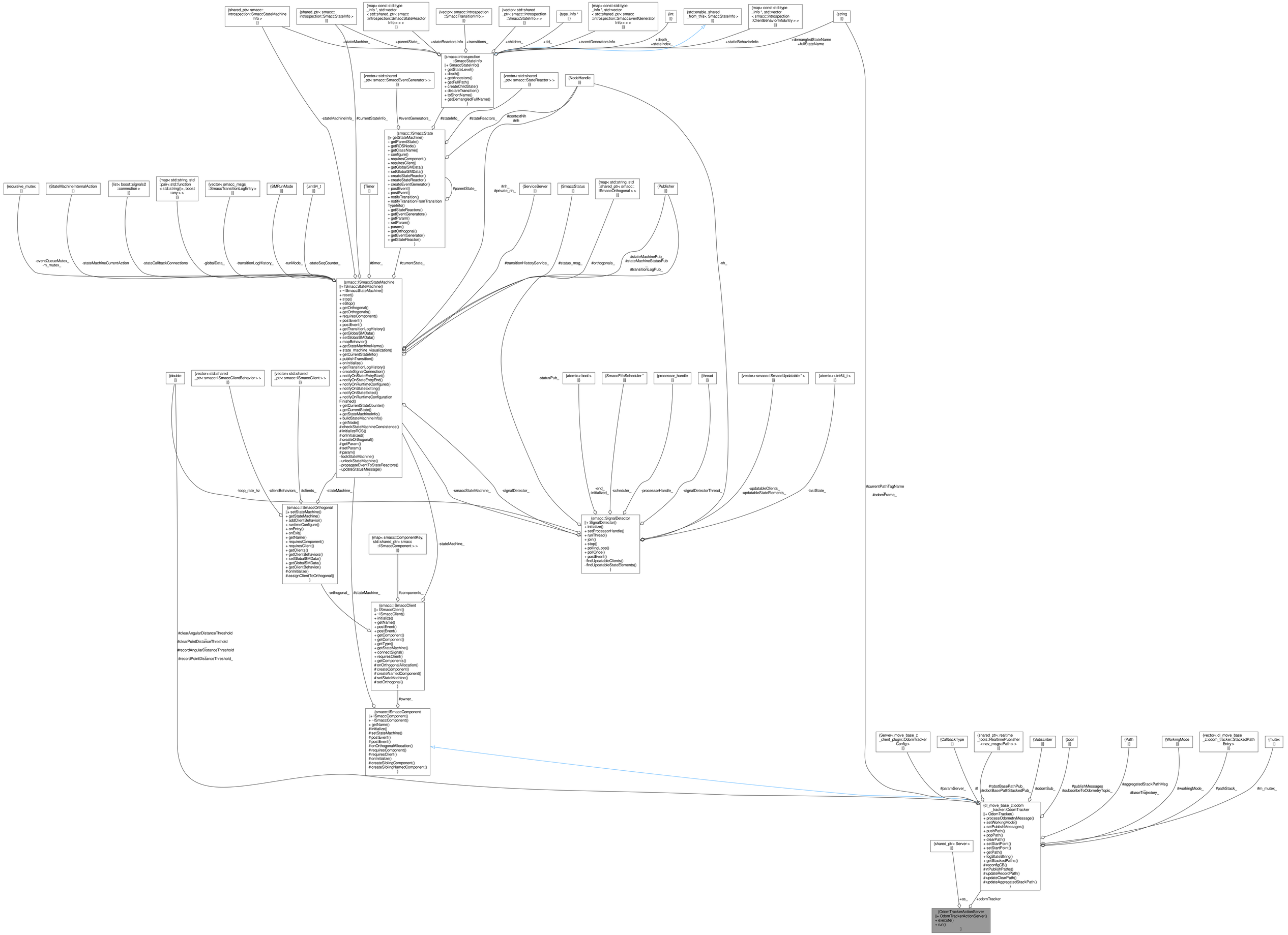
{shared_ptr< smacc::introspection::SmaccStateMachineInfo > }}
{shared_ptr< smacc::introspection::SmaccStateInfo > }}
{map< const std::type_info *, std::vector < std::shared_ptr< smacc::introspection::SmaccStateReactorInfo > > > }}
{vector< smacc::introspection::SmaccTransitionInfo > }}
{vector< std::shared _ptr< smacc::introspection::SmaccStateInfo > > }}
{type_info * }}
{map< const std::type_info *, std::vector < std::shared_ptr< smacc::introspection::SmaccEventGenerator Info > > > }}
{int }}
{std::enable_shared _from_this< SmaccStateInfo > }}
{map< const std::type_info *, std::vector < smacc::introspection::ClientBehaviorInfoEntry > > }}
{string }}
+stateMachine_
+parentState_
+stateReactorsInfo
+transitions_
+children_
+tid_
+eventGeneratorsInfo
+depth_
+stateIndex_
+staticBehaviorInfo
+demangledStateName
+fullStateName
{smacc::introspection ::SmaccStateInfo }|+ SmaccStateInfo() + getStateLevel() + depth() + getAncestors() + getFullPath() + createChildState() + declareTransition() + toShortName() + getDemangledFullName() }
{vector< std::shared _ptr< smacc::SmaccEventGenerator > > }}
{vector< std::shared _ptr< smacc::StateReactor > > }}
{NodeHandle }}
-stateMachineInfo_
#currentStateInfo_
#eventGenerators_
#stateInfo_
#stateReactors_
#contextNh
#nh
{smacc::ISmaccState }|+ getStateMachine() + getParentState() + getROSNode() + getClassName() + configure() + requiresComponent() + requiresClient() + getGlobalSMData() + setGlobalSMData() + createStateReactor() + createStateReactor() + createEventGenerator() + postEvent() + postEvent() + notifyTransition() + notifyTransitionFromTransition TypeInfo() + getStateReactors() + getEventGenerators() + getParam() + setParam() + param() + getOrthogonal() + getEventGenerator() + getStateReactor() }
#parentState_
{recursive_mutex }}
{StateMachineInternalAction }}
{list< boost::signals2 ::connection > }}
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > }}
{vector< smacc_msgs ::SmaccTransitionLogEntry > }}
{SMRunMode }}
{uint64_t }}
{Timer }}
#nh_
#private_nh_
{ServiceServer }}
{SmaccStatus }}
{map< std::string, std ::shared_ptr< smacc:: ISmaccOrthogonal > > }}
{Publisher }}
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
-stateSeqCounter_
#timer_
#currentState_
#transitionHistoryService_
#status_msg_
#orthogonals_
#stateMachinePub_
#stateMachineStatusPub
#transitionLogPub_
-nh_
{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() + notifyOnStateEntryStart() + notifyOnStateEntryEnd() + notifyOnRuntimeConfigured() + notifyOnStateExitting() + notifyOnStateExited() + notifyOnRuntimeConfiguration Finished() + getCurrentStateCounter() + getCurrentState() + getStateMachineInfo() + buildStateMachineInfo() + getNode() # checkStateMachineConsistence() # initializeROS() # onInitialized() # createOrthogonal() # getParam() # setParam() # param() - lockStateMachine() - unlockStateMachine() - propagateEventToStateReactors() - updateStatusMessage() }
{double }}
{vector< std::shared _ptr< smacc::ISmaccClientBehavior > > }}
{vector< std::shared _ptr< smacc::ISmaccClient > > }}
-statusPub_
{atomic< bool > }}
{SmaccFifoScheduler * }}
{processor_handle }}
{thread }}
{vector< smacc::ISmaccUpdatable * > }}
{atomic< uint64_t > }}
-loop_rate_hz
-clientBehaviors_
#clients_
-stateMachine_
-smaccStateMachine_
-signalDetector_
-end_ -initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-updatableClients_ -updatableStateElements_
-lastState_
#currentPathTagName
#odomFrame_
{smacc::ISmaccOrthogonal }|+ setStateMachine() + getStateMachine() + addClientBehavior() + runtimeConfigure() + onEntry() + onExit() + getName() + requiresComponent() + requiresClient() + getClients() + getClientBehaviors() + setGlobalSMData() + getGlobalSMData() + getClientBehavior() # onInitialize() # assignClientToOrthogonal() }
{smacc::ISignalDetector }|+ SignalDetector() + initialize() + setProcessorHandle() + runThread() + join() + stop() + pollingLoop() + pollOnce() + postEvent() - findUpdatableClients() - findUpdatableStateElements() }
{map< smacc::ComponentKey, std::shared_ptr< smacc ::ISmaccComponent > > }}
-stateMachine_
-orthogonal_
#stateMachine_
#components_
{smacc::ISmaccClient }|+ ISmaccClient() + ~ISmaccClient() + initialize() + getName() + postEvent() + postEvent() + getComponent() + getComponent() + getType() + getStateMachine() + connectSignal() + requiresClient() + getComponents() # onOrthogonalAllocation() # createComponent() # createNamedComponent() # setStateMachine() # setOrthogonal() }
#owner_
{smacc::ISmaccComponent }|+ ISmaccComponent() + ~ISmaccComponent() + getName() # initialize() # setStateMachine() # postEvent() # postEvent() # onOrthogonalAllocation() # requiresComponent() # requiresClient() # onInitialize() # createSiblingComponent() # createSiblingNamedComponent() }
#clearAngularDistanceThreshold
#clearPointDistanceThreshold
#recordAngularDistanceThreshold
#recordPointDistanceThreshold_
{Server< move_base_z _client_plugin::OdomTracker Config > }}
{CallbackType }}
{shared_ptr< realtime _tools::RealtimePublisher < nav_msgs::Path > > }}
{Subscriber }}
{bool }}
{Path }}
{WorkingMode }}
{vector< cl_move_base _z::odom_tracker::StackedPath Entry > }}
{mutex }}
#paramServer_
#
#robotBasePathPub_ #robotBasePathStackedPub_
#odomSub_
#publishMessages #subscribeToOdometryTopic_
#aggregatedStackPathMsg
#baseTrajectory_
#workingMode_
#pathStack_
#m_mutex_
{cl_move_base_z::odom _tracker::OdomTracker }|+ OdomTracker() + processOdometryMessage() + setWorkingMode() + setPublishMessages() + pushPath() + popPath() + clearPath() + setStartPoint() + setStartPoint() + getPath() + logStateString() + getStackedPaths() # reconfigCB() # rtPublishPaths() # updateRecordPath() # updateClearPath() # updateAggregatedStackPath() }
{shared_ptr< Server > }}
+as_
+odomTracker
{OdomTrackerActionServer }|+ OdomTrackerActionServer() + execute() + run() }