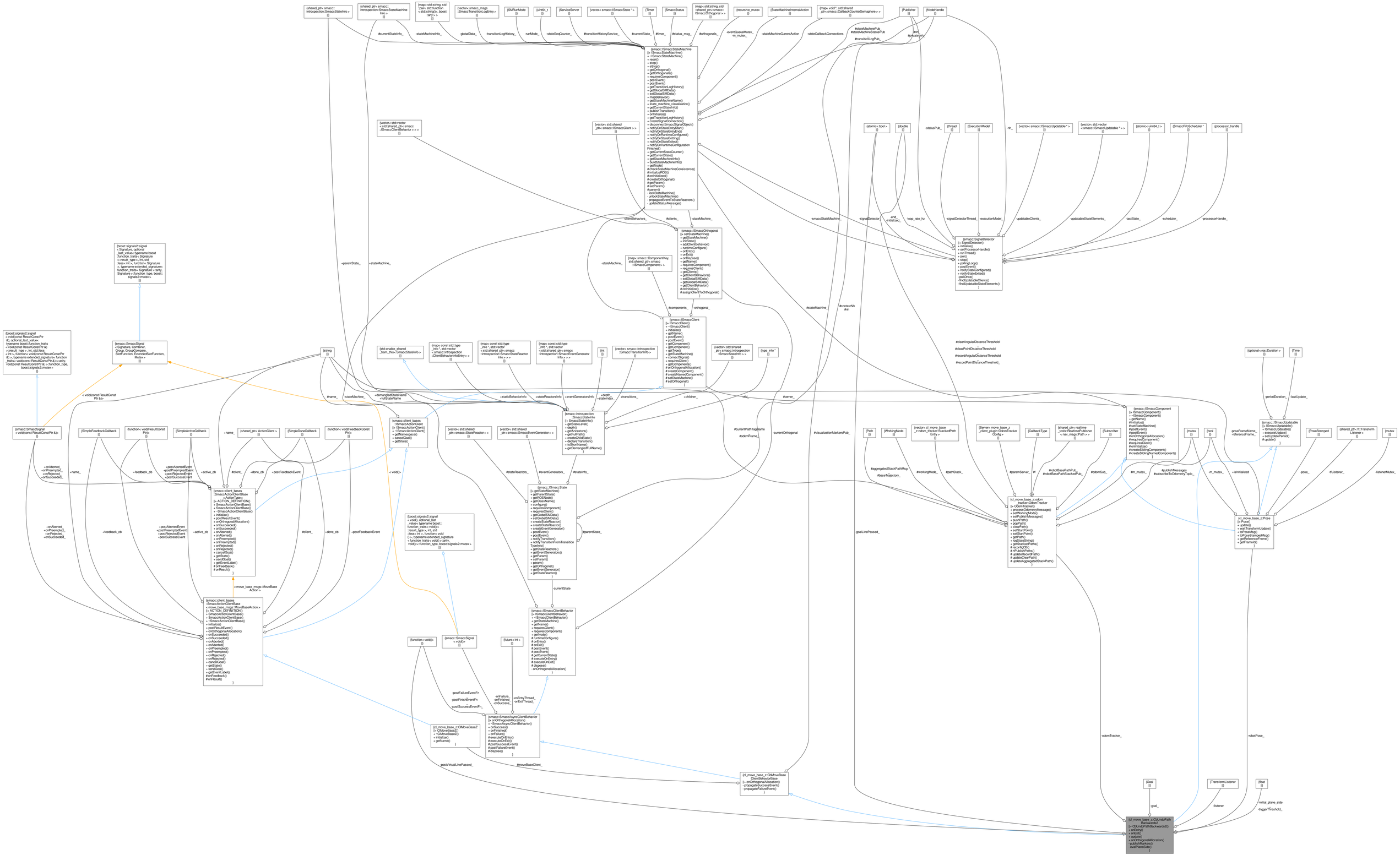
[smacc::ISmaccStateMachine
[= ISmaccStateMachine]
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvents()
+ getTransitionLogHistory()
+ getGlobalSMData()
+ setGlobalSMData()
+ mapBehavior()
+ getStateMachineName()
+ state_machine_visualization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnections()
+ notifyOnStateMachineCreated()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExitting()
+ notifyOnStateExited()
+ notifyOnRuntimeConfiguration
Finished()
+ getCurrentStateCounter()
+ getStateMap()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
# getParam()
# getParam()
# getInt()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
[smacc::ISmaccOrthogonal
[= setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClients()
+ getClients()
+ getClientBehaviors()
+ getClientBehavior()
+ getGlobalSMData()
+ getClientBehavior()
# oninitialize()
# assignClientToOrthogonal()
}
[smacc::ISmaccClient
[= ISmaccClient]
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ getClients()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# onOrthogonalAllocation()
# onOrthogonalExit()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
}
[smacc::ISmaccSignal
< Signature, Combiner,
Group, GroupCompare,
SlotFunction, ExtendedSlotFunction,
Mutex >
]
[smacc::client_bases
:SmaccActionClientBase
< ActionType >
[= ACTION_DEFINITION]
+ SmaccActionClientBase()
+ ~SmaccActionClientBase()
+ initialize()
+ postResultEvent()
+ onOrthogonalAllocation()
+ onSucceeded()
+ onSucceeded()
+ onAborted()
+ onAborted()
+ onPreempted()
+ onPreempted()
+ onRejected()
+ onRejected()
+ cancelGoal()
+ getState()
+ sendGoal()
+ getEventLabel()
# onFeedback()
# onResult()
}
[smacc::client_bases
:SmaccActionClientBase
< move_base_msgs::MoveBaseAction >
[= ACTION_DEFINITION]
+ SmaccActionClientBase()
+ SmaccActionClientBase()
+ ~SmaccActionClientBase()
+ initialize()
+ postResultEvent()
+ onOrthogonalAllocation()
+ onSucceeded()
+ onSucceeded()
+ onAborted()
+ onAborted()
+ onPreempted()
+ onPreempted()
+ onRejected()
+ onRejected()
+ cancelGoal()
+ getState()
+ sendGoal()
+ getEventLabel()
# onFeedback()
# onResult()
}
[smacc::client_bases
:SmaccActionClientBase
[= SmaccActionClient]
+ ~ISmaccActionClient()
+ getNamespace()
+ cancelGoal()
+ getState()
}
[smacc::ISmaccState
[= getStateMachine()
+ getParentState()
+ getROSNode()
+ getStateName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition
TypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getParam()
+ setParam()
+ getInt()
+ getOrthogonal()
+ getEventGenerator()
+ getStateReactor()
}
[smacc::ISmaccClientBehavior
[= ISmaccClientBehavior]
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ setState()
+ requiresClient()
+ requiresComponent()
+ getNode()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# postEventToState()
# executeOnEntry()
# executeOnExit()
# dispose()
- onOrthogonalAllocation()
}
[smacc::SmaccAsyncClientBehavior
[= onOrthogonalAllocation()]
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
# executeOnEntry()
# executeOnExit()
# postSuccessEvent()
# postFailureEvent()
# dispose()
}
[cl_move_base_z:CtMoveBaseZ
[= CtMoveBaseZ]
+ ~CtMoveBaseZ()
+ initialize()
+ getName()
}
[cl_move_base_z:CtMoveBase
:ClientBehaviorBase
[= onOrthogonalAllocation()]
- propagateSuccessEvent()
- propagateFailureEvent()
}
[smacc::ISmaccComponent
[= ISmaccComponent]
+ ~ISmaccComponent()
+ getName()
# initialize()
# setStateMachine()
# postEvent()
# postEvent()
# onOrthogonalAllocation()
# requiresClient()
# requiresComponent()
# oninitialize()
# createSiblingComponent()
# createSiblingNamedComponent()
}
[smacc::ISmaccUpdatable
[= ISmaccUpdatable]
+ ~ISmaccUpdatable()
+ execute(Update)
+ setUpdatePeriod()
# update()
}
[cl_move_base_z:CtIndexPath
Backwards2
[= CtIndexPathBackwards2]
+ onEntry()
+ onExit()
+ update()
- onOrthogonalAllocation()
- publishMarkers()
- evalPlaneSide()
}
[cl_move_base_z:odom
_tracker:OdomTracker
[= OdomTracker]
+ processOdometryMessage()
+ setWorkingMode()
+ setWorkMessages()
+ setPath()
+ popPath()
+ clearPath()
+ setStartPoint()
+ setStartPoint()
+ getPath()
+ logStateString()
+ getStackedPaths()
+ publishPath()
# publishPaths()
# updateRecordPath()
# updateClearPath()
# updateAggregatedStackPath()
}
[cl_move_base_z:Pose
[= Pose]
+ update()
+ waitTransformUpdate()
+ getPoseMsg()
+ toPoseStampedMsg()
+ getReferenceFrame()
+ getFrameId()
}
[smacc::SignalDetector
[= SignalDetector]
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ postEvent-oop()
+ postEvent()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClients()
- findUpdatableStateElements()
}