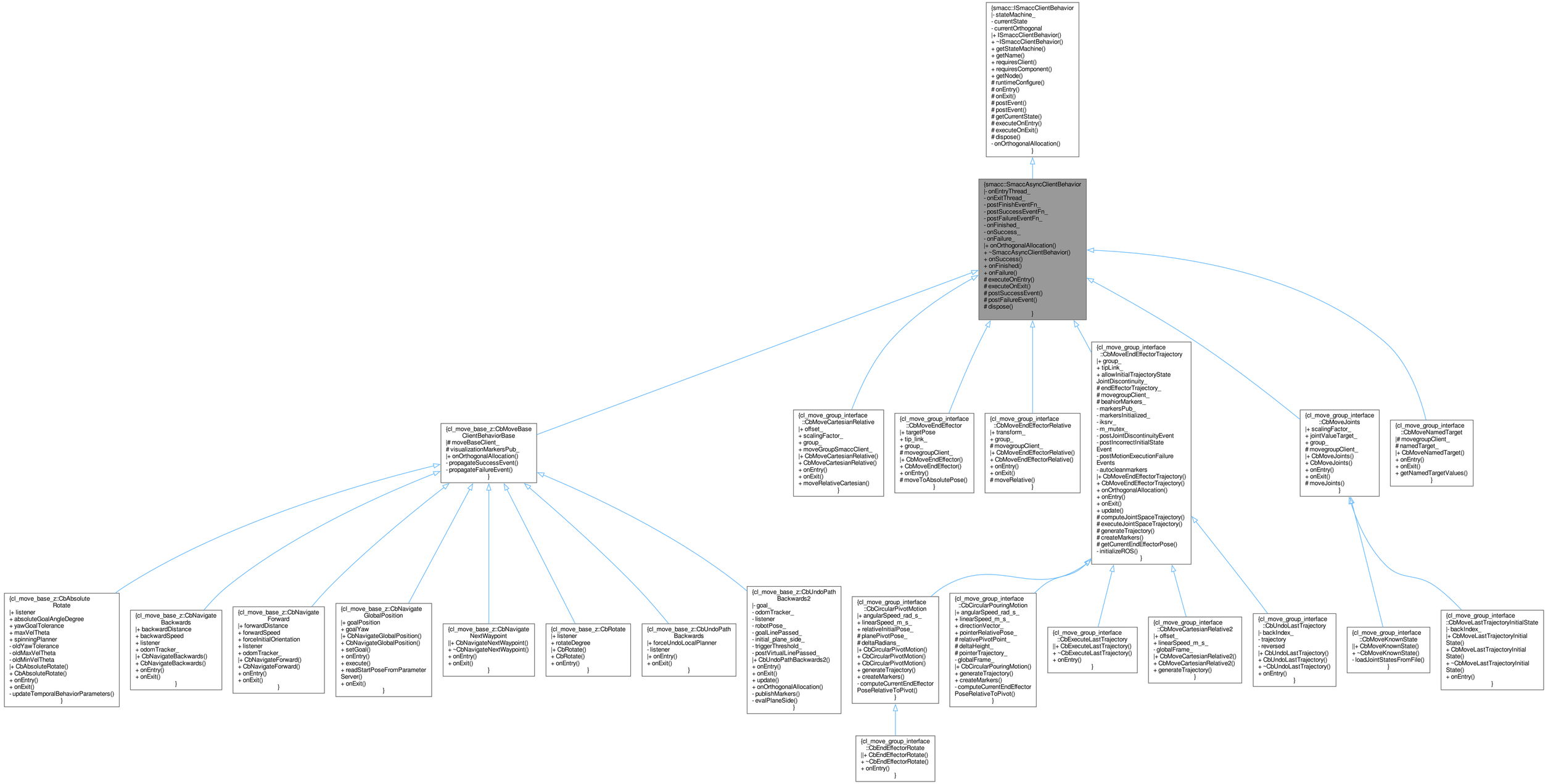
{smacc::ISmaccClientBehavior
|- stateMachine_
- currentState
- currentOrthogonal
|+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ getNode()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# getCurrentState()
# executeOnEntry()
# executeOnExit()
# dispose()
- onOrthogonalAllocation()
}
{smacc::SmaccAsyncClientBehavior
|- onEntryThread_
- onExitThread_
- postFinishEventFn_
- postSuccessEventFn_
- postFailureEventFn_
- onFinished_
- onSuccess_
- onFailure_
|+ onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
# executeOnEntry()
# executeOnExit()
# postSuccessEvent()
# postFailureEvent()
# dispose()
}
{cl_move_base_z::CbMoveBase
ClientBehaviorBase
|# moveBaseClient_
# visualizationMarkersPub_
|+ onOrthogonalAllocation()
- propagateSuccessEvent()
- propagateFailureEvent()
}
{cl_move_group_interface
::CbMoveCartesianRelative
|+ offset_
+ scalingFactor_
+ group_
+ moveGroupSmaccClient_
|+ CbMoveCartesianRelative()
+ CbMoveCartesianRelative()
+ onEntry()
+ onExit()
+ moveRelativeCartesian()
}
{cl_move_group_interface
::CbMoveEndEffector
|+ targetPose
+ tip_link_
+ group_
# movegroupClient_
|+ CbMoveEndEffector()
+ CbMoveEndEffector()
+ onEntry()
# moveToAbsolutePose()
}
{cl_move_group_interface
::CbMoveEndEffectorRelative
|+ transform_
+ group_
# movegroupClient_
|+ CbMoveEndEffectorRelative()
+ CbMoveEndEffectorRelative()
+ onEntry()
+ onExit()
# moveRelative()
}
{cl_move_group_interface
::CbMoveEndEffectorTrajectory
|+ group_
+ tipLink_
+ allowInitialTrajectoryState
JointDiscontinuity_
# endEffectorTrajectory_
# movegroupClient_
# beahiorMarkers_
- markersPub_
- markersInitialized_
- iksrv_
- m_mutex_
- postJointDiscontinuityEvent
- postIncorrectInitialState
Event
- postMotionExecutionFailure
Events
- autocleanmarkers
|+ CbMoveEndEffectorTrajectory()
+ CbMoveEndEffectorTrajectory()
+ onOrthogonalAllocation()
+ onEntry()
+ onExit()
+ update()
# computeJointSpaceTrajectory()
# executeJointSpaceTrajectory()
# generateTrajectory()
# createMarkers()
# getCurrentEndEffectorPose()
- initializeROS()
}
{cl_move_group_interface
::CbMoveJoints
|+ scalingFactor_
+ jointValueTarget_
+ group_
# movegroupClient_
|+ CbMoveJoints()
+ CbMoveJoints()
+ onEntry()
+ onExit()
# moveJoints()
}
{cl_move_group_interface
::CbMoveNamedTarget
|# movegroupClient_
# namedTarget_
|+ CbMoveNamedTarget()
+ onEntry()
+ onExit()
+ getNamedTargetValues()
}
{cl_move_base_z::CbAbsolute
Rotate
|+ listener
+ absoluteGoalAngleDegree
+ yawGoalTolerance
+ maxVelTheta
+ spinningPlanner
- oldYawTolerance
- oldMaxVelTheta
- oldMinVelTheta
|+ CbAbsoluteRotate()
+ CbAbsoluteRotate()
+ onEntry()
+ onExit()
- updateTemporalBehaviorParameters()
}
{cl_move_base_z::CbNavigate
Backwards
|+ backwardDistance
+ backwardSpeed
+ listener
+ odomTracker_
|+ CbNavigateBackwards()
+ CbNavigateBackwards()
+ onEntry()
+ onExit()
}
{cl_move_base_z::CbNavigate
Forward
|+ forwardDistance
+ forwardSpeed
+ forceInitialOrientation
+ listener
+ odomTracker_
|+ CbNavigateForward()
+ CbNavigateForward()
+ onEntry()
+ onExit()
}
{cl_move_base_z::CbNavigate
GlobalPosition
|+ goalPosition
+ goalYaw
|+ CbNavigateGlobalPosition()
+ CbNavigateGlobalPosition()
+ setGoal()
+ onEntry()
+ execute()
+ readStartPoseFromParameter
Server()
+ onExit()
}
{cl_move_base_z::CbNavigate
NextWaypoint
||+ CbNavigateNextWaypoint()
+ ~CbNavigateNextWaypoint()
+ onEntry()
+ onExit()
}
{cl_move_base_z::CbRotate
|+ listener
+ rotateDegree
|+ CbRotate()
+ CbRotate()
+ onEntry()
}
{cl_move_base_z::CbUndoPath
Backwards
|+ forceUndoLocalPlanner
- listener
|+ onEntry()
+ onExit()
}
{cl_move_base_z::CbUndoPath
Backwards2
|- goal_
- odomTracker_
- listener
- robotPose_
- goalLinePassed_
- initial_plane_side_
- triggerThreshold_
- postVirtualLinePassed_
|+ CbUndoPathBackwards2()
+ onEntry()
+ onExit()
+ update()
+ onOrthogonalAllocation()
- publishMarkers()
- evalPlaneSide()
}
{cl_move_group_interface
::CbCircularPivotMotion
|+ angularSpeed_rad_s_
+ linearSpeed_m_s_
+ relativeInitialPose_
# planePivotPose_
# deltaRadians_
|+ CbCircularPivotMotion()
+ CbCircularPivotMotion()
+ CbCircularPivotMotion()
+ generateTrajectory()
+ createMarkers()
- computeCurrentEndEffector
PoseRelativeToPivot()
}
{cl_move_group_interface
::CbCircularPouringMotion
|+ angularSpeed_rad_s_
+ linearSpeed_m_s_
+ directionVector_
+ pointerRelativePose_
# relativePivotPoint_
# deltaHeight_
# pointerTrajectory_
- globalFrame_
|+ CbCircularPouringMotion()
+ generateTrajectory()
+ createMarkers()
- computeCurrentEndEffector
PoseRelativeToPivot()
}
{cl_move_group_interface
::CbEndEffectorRotate
||+ CbEndEffectorRotate()
+ ~CbEndEffectorRotate()
+ onEntry()
}
{cl_move_group_interface
::CbExecuteLastTrajectory
||+ CbExecuteLastTrajectory()
+ ~CbExecuteLastTrajectory()
+ onEntry()
}
{cl_move_group_interface
::CbMoveCartesianRelative2
|+ offset_
+ linearSpeed_m_s_
- globalFrame_
|+ CbMoveCartesianRelative2()
+ CbMoveCartesianRelative2()
+ generateTrajectory()
}
{cl_move_group_interface
::CbUndoLastTrajectory
|- backIndex_
- trajectory
- reversed
|+ CbUndoLastTrajectory()
+ CbUndoLastTrajectory()
+ ~CbUndoLastTrajectory()
+ onEntry()
}
{cl_move_group_interface
::CbMoveKnownState
||+ CbMoveKnownState()
+ ~CbMoveKnownState()
- loadJointStatesFromFile()
}
{cl_move_group_interface
::CbMoveLastTrajectoryInitialState
|- backIndex_
|+ CbMoveLastTrajectoryInitial
State()
+ CbMoveLastTrajectoryInitial
State()
+ ~CbMoveLastTrajectoryInitial
State()
+ onEntry()
}