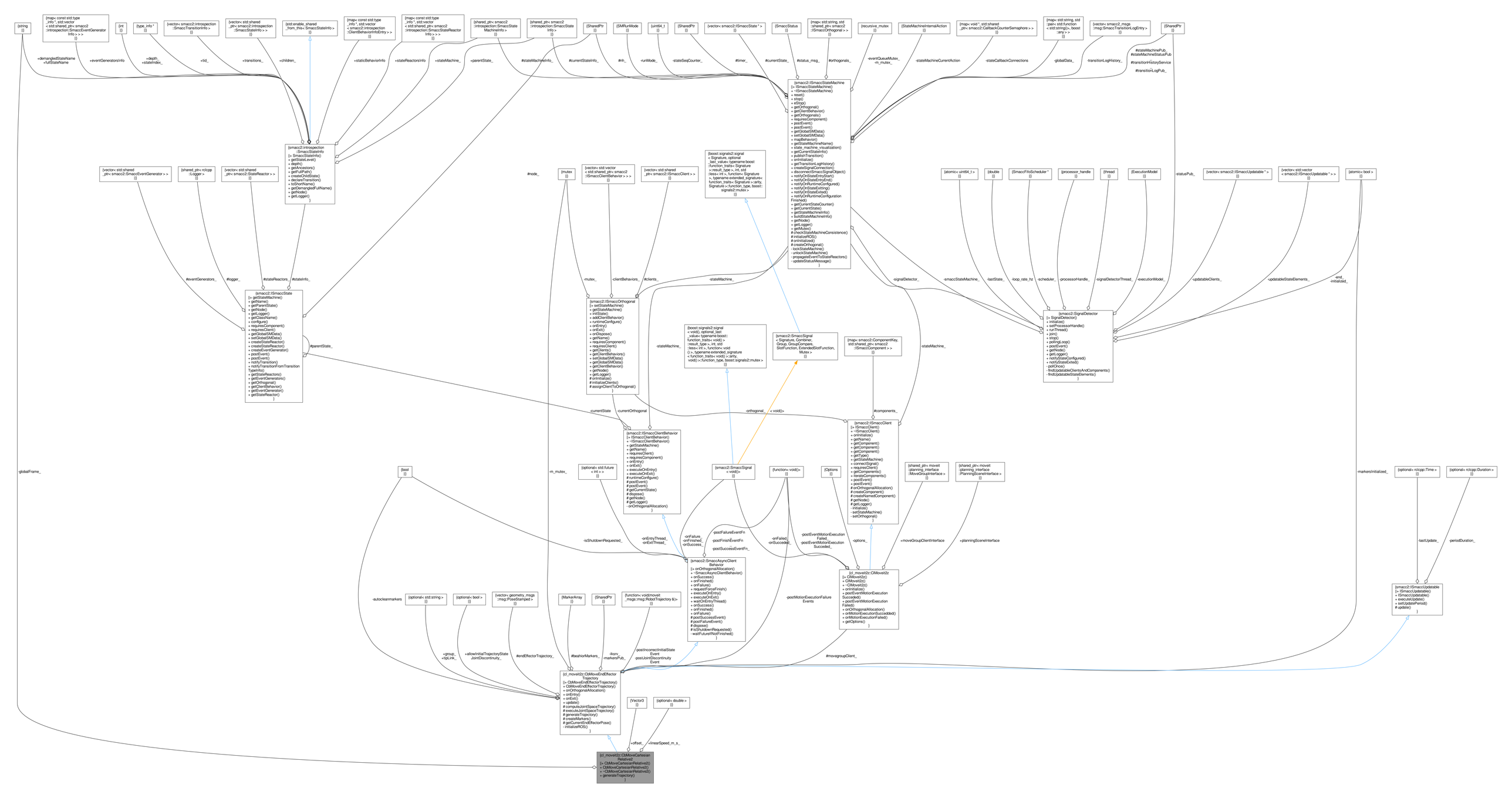
[smacc2::ISmaccStateMachine
[+ ISmaccStateMachine()
~ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getClientBehavior()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ getGlobalSMData()
+ getGlobalSMData()
+ mapBehavior()
+ getStateMachineName()
+ state_machine_initialization()
+ getCurrentStateInfo()
+ publishTransition()
+ onInitialize()
+ getTransitionLogHistory()
+ createSignalConnection()
+ disconnectSmaccSignalObject()
+ notifyOnStateEntryStart()
+ notifyOnStateEntryEnd()
+ notifyOnRuntimeConfigured()
+ notifyOnStateExiting()
+ notifyOnStateExited()
+ notifyOnRuntimeConfiguration
Finished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
+ getLogger()
+ getMutex()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
[smacc2::ISmaccSignal
< Signature, Combiner,
Group, GroupCompare,
SlotFunction, ExtendedSlotFunction,
Mutex >
]]
[smacc2::ISmaccClient
[+ ISmaccClient()
+ ~ ~ISmaccClient()
+ onInitialize()
+ getName()
+ getComponent()
+ getComponent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
+ iterateComponents()
+ postEvent()
+ postEvent()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# getNode()
# getLogger()
- initialize()
- setStateMachine()
- setOrthogonal()
}
[smacc2::SignalDetector
[+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
}
[smacc2::ISmaccUpdatable
[+ ISmaccUpdatable()
+ ISmaccUpdatable()
+ execute(Update)
+ setUpdatePeriod()
# update()
}
[cl_moveit2z::CtMoveit2z
[+ CtMoveit2z()
+ CtMoveit2z()
+ ~CtMoveit2z()
+ onInitialize()
+ postEventMotionExecution
Succeeded()
+ postEventMotionExecution
Failed()
+ onOrthogonalAllocation()
+ onMotionExecutionSucceeded()
+ onMotionExecutionFailed()
+ getOptions()
}
[smacc2::SmaccAsyncClient
Behavior
[+ onOrthogonalAllocation()
+ ~SmaccAsyncClientBehavior()
+ onSuccess()
+ onFinished()
+ onFailure()
+ requestForceFinish()
+ executeOnEntry()
+ executeOnExit()
+ waitOnEntryThread()
+ onSuccess()
+ onFinished()
+ onFailure()
# postSuccessEvent()
# postFailureEvent()
# dispose()
# isShutdownRequested()
- waitFutureIfNotFinished()
}
[smacc2::ISmaccClientBehavior
[+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ onEntry()
+ onExit()
+ executeOnEntry()
+ executeOnExit()
# runtimeConfigure()
# postEvent()
# postEvent()
# getCurrentState()
# dispose()
# getNode()
# getLogger()
- onOrthogonalAllocation()
}
[smacc2::ISmaccOrthogonal
[+ setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
# onInitialize()
# initializeClients()
# assignClientToOrthogonal()
}
[smacc2::ISmaccState
[+ getStateMachine()
+ getName()
+ getParentState()
+ getNode()
+ getLogger()
+ getStateName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransition
TypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getOrthogonal()
+ getClientBehavior()
+ getEventGenerator()
+ getStateReactor()
}
[smacc2::introspection
::SmaccStateInfo
[+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
+ getNode()
+ getLogger()
}
[cl_movebt2z::CtMoveEndEffector
Trajectory
[+ CtMoveEndEffectorTrajectory()
+ ~CtMoveEndEffectorTrajectory()
+ onOrthogonalAllocation()
+ onEntry()
+ onExit()
+ update()
# computeJointSpaceTrajectory()
# executeJointSpaceTrajectory()
# generateTrajectory()
# createMarkers()
# getCurrentEndEffectorPose()
- initializeROS()
}
[cl_movebt2z::CtMoveCartesian
Relative2
[+ CtMoveCartesianRelative2()
+ ~CtMoveCartesianRelative2()
+ <CtMoveCartesianRelative2()
+ generateTrajectory()
}
[string
]]
[map< const std::type
_info *, std::vector
< smacc2::introspection
::SmaccEventGenerator
Info > > >
]]
[int
]]
[type_info *
]]
[vector< smacc2::introspection
::SmaccTransitionInfo >
]]
[vector< std::shared
_ptr< smacc2::introspection
::SmaccStateInfo > >
]]
[std::enable_shared
_from_this< SmaccStateInfo >
]]
[map< const std::type
_info *, std::vector
< smacc2::introspection
::ClientBehaviorInfoEntry > >
]]
[map< const std::type
_info *, std::vector
< std::shared_ptr< smacc2
::introspection::SmaccStateReactor
Info > > >
]]
[shared_ptr< smacc2
::introspection::SmaccState
MachineInfo >
]]
[shared_ptr< smacc2
::introspection::SmaccState
Info >
]]
[SharedPtr
]]
[SMRunMode
]]
[uint64_t
]]
[SharedPtr
]]
[vector< smacc2::ISmaccState * >
]]
[SmaccStatus
]]
[map< std::string, std
::shared_ptr< smacc2
::ISmaccOrthogonal > >
]]
[recursive_mutex
]]
[StateMachineInternalAction
]]
[map< void *, std::shared
_ptr< smacc2::CallbackCounterSemaphore > >
]]
[map< std::string, std
::pair< std::function
< std::string()>, boost
::any > >
]]
[vector< smacc2_msgs
::msg::SmaccTransitionLogEntry >
]]
[SharedPtr
]]
[vector< std::shared
_ptr< smacc2::SmaccEventGenerator > >
]]
[shared_ptr< rclcpp
::Logger >
]]
[vector< std::shared
_ptr< smacc2::StateReactor > >
]]
[vector< std::vector
< std::shared_ptr< smacc2
::ISmaccClientBehavior > > >
]]
[vector< std::shared
_ptr< smacc2::ISmaccClient > >
]]
[boost::signals2::signal
< Signature, optional
_last_value= typename boost
::function_traits< Signature
>::result_type >, int, std
::less< int >, function< Signature
>, typename extended_signature<
function_traits< Signature >::arity,
Signature >::function_type, boost::
signals2::mutex >
]]
[atomic< uint64_t >
]]
[double
]]
[SmaccFifoScheduler *
]]
[processor_handle
]]
[thread
]]
[ExecutionModel
]]
[vector< smacc2::ISmaccUpdatable * >
]]
[vector< std::vector
< smacc2::ISmaccUpdatable * > >
]]
[atomic< bool >
]]
[mutex
]]
[boost::signals2::signal
< void(), optional_last
_value= typename boost::
function_traits< void() >
::result_type >, int, std
::less< int >, function< void
() >, typename extended_signature
< function_traits< void() >::arity,
void() >::function_type, boost::signals2::mutex >
]]
[map< smacc2::ComponentKey,
std::shared_ptr< smacc2
::ISmaccComponent > >
]]
[smacc2::SmaccSignal
< void()>
]]
[function< void()>
]]
[Options
]]
[shared_ptr< moveit
::planning_interface
::MoveGroupInterface >
]]
[shared_ptr< moveit
::planning_interface
::PlanningSceneInterface >
]]
[optional< std::future
< int > >
]]
[bool
]]
[optional< std::string >
]]
[optional< bool >
]]
[vector< geometry_msgs
::msg::PoseStamped >
]]
[MarkerArray
]]
[SharedPtr
]]
[function< void(moveit
::msgs::msg::RobotTrajectory &)>
]]
[Vector3
]]
[optional< double >
]]
[optional< rclcpp::Time >
]]
[optional< rclcpp::Duration >
]]
#node_
-mutex_
-clientBehaviors_
#clients_
-stateMachine_
-signalDetector_
-smaccStateMachine_
-lastState_
-loop_rate_hz_
-scheduler_
processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-end_
-initialized_
-statusPub_
#stateMachinePub_
#stateMachineStatusPub
#transitionHistoryService
#transitionLogPub_
-globalFrame_
-eventGenerators_
#logger_
#stateReactors_
#stateInfo_
#parentState_
-currentState
-currentOrthogonal
-orthogonal_
< void()>
#components_
-m_mutex_
-markersInitialized_
-lastUpdate_
-periodDuration_
-isShutdownRequested_
-onEntryThread_
-onExitThread_
-onFailure_
-onFinished_
-onSuccess_
-postFailureEventFn
-postFinishEventFn
-postSuccessEventFn_
-onFailed_
-onSucceeded_
-postEventMotionExecution
Failed_
-postEventMotionExecution
Succeeded_
-options_
+moveGroupClientInterface
+planningSceneInterface
-autocleanmarkers
+group_
+tipLink_
+allowInitialTrajectoryState
JointDiscontinuity_
#endEffectorTrajectory_
#beahiorMarkers_
-iksrv_
-markersPub_
-postIncorrectInitialState
Event
-postJointDiscontinuity
Event
-postMotionExecutionFailure
Events
#movegroupClient_
+offset_
+linearSpeed_m_s_