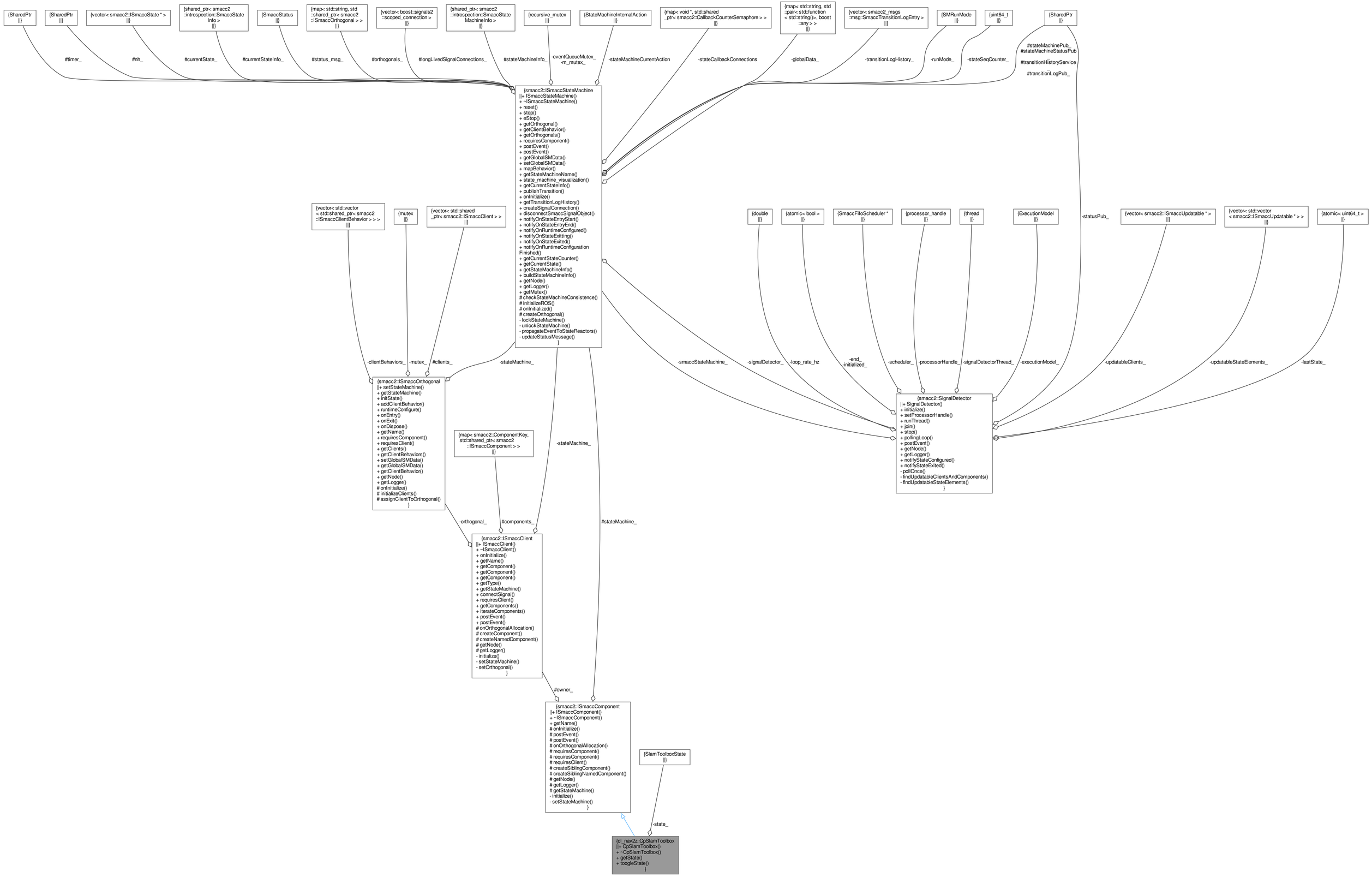
{SharedPtr
|}}
#timer_
{SharedPtr
|}}
#nh_
{vector< smacc2::ISmaccState * >
|}}
#currentState_
{shared_ptr< smacc2
::introspection::SmaccState
Info >
|}}
#currentStateInfo_
{SmaccStatus
|}}
#status_msg_
{map< std::string, std
::shared_ptr< smacc2
::ISmaccOrthogonal > >
|}}
#orthogonals_
{vector< boost::signals2
::scoped_connection >
|}}
#longLivedSignalConnections_
{shared_ptr< smacc2
::introspection::SmaccState
MachineInfo >
|}}
#stateMachineInfo_
{recursive_mutex
|}}
-eventQueueMutex_
-m_mutex_
{StateMachineInternalAction
|}}
-stateMachineCurrentAction
{map< void *, std::shared
_ptr< smacc2::CallbackCounterSemaphore > >
|}}
-stateCallbackConnections
{map< std::string, std
::pair< std::function
< std::string()>, boost
::any > >
|}}
-globalData_
{vector< smacc2_msgs
::msg::SmaccTransitionLogEntry >
|}}
-transitionLogHistory_
{SMRunMode
|}}
-runMode_
{uint64_t
|}}
-stateSeqCounter_
{SharedPtr
|}}
#stateMachinePub_
#stateMachineStatusPub
_
#transitionHistoryService
_
#transitionLogPub_
{smacc2::ISmaccStateMachine
||+ ISmaccStateMachine()
+ ~ISmaccStateMachine()
+ reset()
+ stop()
+ eStop()
+ getOrthogonal()
+ getClientBehavior()
+ getOrthogonals()
+ requiresComponent()
+ postEvent()
+ postEvent()
+ 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()
+ getLogger()
+ getMutex()
# checkStateMachineConsistency()
# initializeROS()
# onInitialized()
# createOrthogonal()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
{vector< std::vector
< std::shared_ptr< smacc2
::ISmaccClientBehavior > > >
|}}
-clientBehaviors_
{mutex
|}}
-mutex_
{vector< std::shared
_ptr< smacc2::ISmaccClient > >
|}}
#clients_
{double
|}}
{atomic< bool >
|}}
{SmaccFifoScheduler *
|}}
{processor_handle
|}}
{thread
|}}
{ExecutionModel
|}}
-statusPub_
{vector< smacc2::ISmaccUpdatable * >
|}}
{vector< std::vector
< smacc2::ISmaccUpdatable * > >
|}}
{atomic< uint64_t >
|}}
-stateMachine_
-smaccStateMachine_
-signalDetector_
-loop_rate_hz
-end_
-initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
{smacc2::ISmaccOrthogonal
||+ setStateMachine()
+ getStateMachine()
+ initState()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ onDispose()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
+ getNode()
+ getLogger()
# onInitialize()
# initializeClients()
# assignClientToOrthogonal()
}
{smacc2::SignalDetector
||+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ postEvent()
+ getNode()
+ getLogger()
+ notifyStateConfigured()
+ notifyStateExited()
- pollOnce()
- findUpdatableClientsAndComponents()
- findUpdatableStateElements()
}
{map< smacc2::ComponentKey,
std::shared_ptr< smacc2
::ISmaccComponent > >
|}}
-stateMachine_
-orthogonal_
#components_
#stateMachine_
{smacc2::ISmaccClient
||+ ISmaccClient()
+ ~ISmaccClient()
+ onInitialize()
+ getName()
+ getComponent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
+ iterateComponents()
+ postEvent()
+ postEvent()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# getNode()
# getLogger()
- initialize()
- setStateMachine()
- setOrthogonal()
}
#owner_
{smacc2::ISmaccComponent
||+ ISmaccComponent()
+ ~ISmaccComponent()
+ getName()
# onInitialize()
# postEvent()
# postEvent()
# onOrthogonalAllocation()
# requiresComponent()
# requiresComponent()
# requiresClient()
# createSiblingComponent()
# createSiblingNamedComponent()
# getNode()
# getLogger()
# getStateMachine()
- initialize()
- setStateMachine()
}
{SlamToolboxState
|}}
-state_
{cl_nav2z::CpSlamToolbox
||+ CpSlamToolbox()
+ ~CpSlamToolbox()
+ getState()
+ toogleState()
}