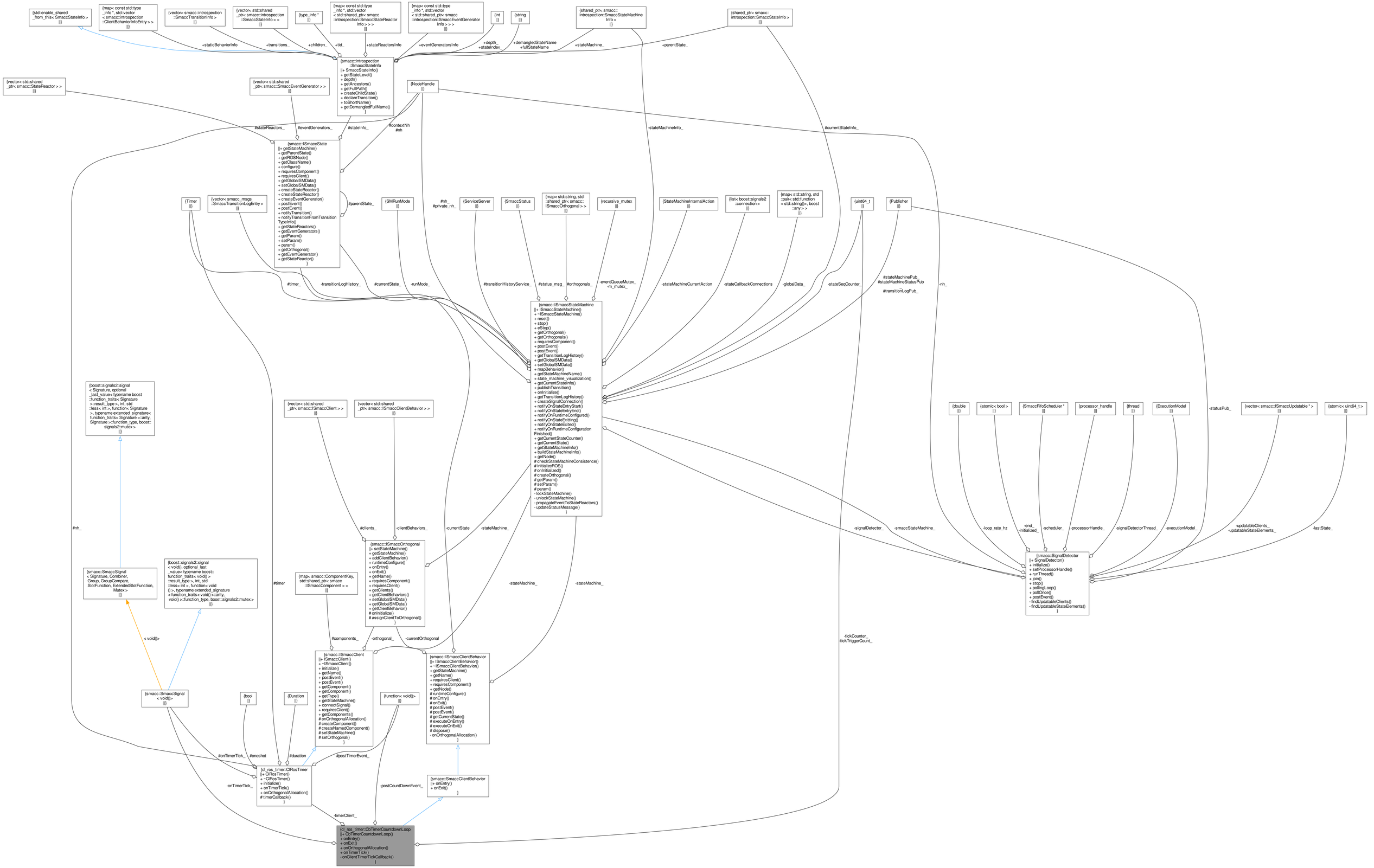
{std::enable_shared_from_this< SmaccStateInfo > ||}
{map< const std::type_info *, std::vector< smacc::introspection::ClientBehaviorInfoEntry > > ||}
{vector< smacc::introspection::SmaccTransitionInfo > ||}
{vector< std::shared_ptr< smacc::introspection::SmaccStateInfo > > ||}
{type_info * ||}
{map< const std::type_info *, std::vector< std::shared_ptr< smacc::introspection::SmaccStateReactorInfo > > > ||}
{map< const std::type_info *, std::vector< std::shared_ptr< smacc::introspection::SmaccEventGeneratorInfo > > > ||}
{int ||}
{string ||}
{shared_ptr< smacc::introspection::SmaccStateMachineInfo > ||}
{shared_ptr< smacc::introspection::SmaccStateInfo > ||}
+staticBehaviorInfo
+transitions_
+children_
+tid_
+stateReactorsInfo
+eventGeneratorsInfo
+depth_
+stateIndex_
+demangledStateName
+fullStateName
+stateMachine_
+parentState_
{smacc::introspection::SmaccStateInfo ||+ SmaccStateInfo()
+ getStateLevel()
+ depth()
+ getAncestors()
+ getFullPath()
+ createChildState()
+ declareTransition()
+ toShortName()
+ getDemangledFullName()
}
{vector< std::shared_ptr< smacc::StateReactor > > ||}
{vector< std::shared_ptr< smacc::SmaccEventGenerator > > ||}
{NodeHandle ||}
#stateReactors_
#eventGenerators_
#stateInfo_
#contextNh
#nh
-stateMachineInfo_
#currentStateInfo_
{smacc::ISmaccState ||+ getStateMachine()
+ getParentState()
+ getROSNode()
+ getClassName()
+ configure()
+ requiresComponent()
+ requiresClient()
+ getGlobalSMData()
+ setGlobalSMData()
+ createStateReactor()
+ createStateReactor()
+ createEventGenerator()
+ postEvent()
+ postEvent()
+ notifyTransition()
+ notifyTransitionFromTransitionTypeInfo()
+ getStateReactors()
+ getEventGenerators()
+ getParam()
+ setParam()
+ param()
+ getOrthogonal()
+ getEventGenerator()
+ getStateReactor()
}
{Timer ||}
{vector< smacc_msgs::SmaccTransitionLogEntry > ||}
#parentState_
{SMRunMode ||}
#nh_
#private_nh_
{ServiceServer ||}
{SmaccStatus ||}
{map< std::string, std::shared_ptr< smacc::ISmaccOrthogonal > > ||}
{recursive_mutex ||}
{StateMachineInternalAction ||}
{list< boost::signals2::connection > ||}
{map< std::string, std::pair< std::function< std::string()>, boost::any > > ||}
{uint64_t ||}
{Publisher ||}
#timer_
-transitionLogHistory_
#currentState_
-runMode_
#transitionHistoryService_
#status_msg_
#orthogonals_
-eventQueueMutex_
-m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-stateSeqCounter_
#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()
+ notifyOnRuntimeConfigurationFinished()
+ getCurrentStateCounter()
+ getCurrentState()
+ getStateMachineInfo()
+ buildStateMachineInfo()
+ getNode()
# checkStateMachineConsistence()
# initializeROS()
# onInitialized()
# createOrthogonal()
# getParam()
# setParam()
# param()
- lockStateMachine()
- unlockStateMachine()
- propagateEventToStateReactors()
- updateStatusMessage()
}
{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 > ||}
{vector< std::shared_ptr< smacc::ISmaccClient > > ||}
{vector< std::shared_ptr< smacc::ISmaccClientBehavior > > ||}
{double ||}
{atomic< bool > ||}
{ISmaccFifoScheduler * ||}
{processor_handle ||}
{thread ||}
{ExecutionModel ||}
-statusPub_
{vector< smacc::ISmaccUpdatable * > ||}
{atomic< uint64_t > ||}
#nh_
#clients_
-clientBehaviors_
-currentState
-stateMachine_
-signalDetector_
-smaccStateMachine_
-loop_rate_hz
-end_
-initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
-updatableClients_
-updatableStateElements_
-lastState_
{smacc::ISmaccOrthogonal ||+ setStateMachine()
+ getStateMachine()
+ addClientBehavior()
+ runtimeConfigure()
+ onEntry()
+ onExit()
+ getName()
+ requiresComponent()
+ requiresClient()
+ getClients()
+ getClientBehaviors()
+ setGlobalSMData()
+ getGlobalSMData()
+ getClientBehavior()
# onInitialize()
# assignClientToOrthogonal()
}
{smacc::SmaccSignal< Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, 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< smacc::ComponentKey, std::shared_ptr< smacc::ISmaccComponent > > ||}
-stateMachine_
-stateMachine_
{smacc::SignalDetector ||+ SignalDetector()
+ initialize()
+ setProcessorHandle()
+ runThread()
+ join()
+ stop()
+ pollingLoop()
+ pollOnce()
+ postEvent()
- findUpdatableClients()
- findUpdatableStateElements()
}
< void()>
#timer
#components_
-orthogonal_
-currentOrthogonal
-tickCounter_
-tickTriggerCount_
{smacc::ISmaccClient ||+ ISmaccClient()
+ ~ISmaccClient()
+ initialize()
+ getName()
+ postEvent()
+ postEvent()
+ getComponent()
+ getComponent()
+ getType()
+ getStateMachine()
+ connectSignal()
+ requiresClient()
+ getComponents()
# onOrthogonalAllocation()
# createComponent()
# createNamedComponent()
# setStateMachine()
# setOrthogonal()
}
{smacc::ISmaccClientBehavior ||+ ISmaccClientBehavior()
+ ~ISmaccClientBehavior()
+ getStateMachine()
+ getName()
+ requiresClient()
+ requiresComponent()
+ getNode()
# runtimeConfigure()
# onEntry()
# onExit()
# postEvent()
# postEvent()
# getCurrentState()
# executeOnEntry()
# executeOnExit()
# dispose()
- onOrthogonalAllocation()
}
{smacc::SmaccSignal< void()> ||}
{bool ||}
{Duration ||}
{function< void()> ||}
#onTimerTick_
#oneshot
#duration
#postTimerEvent_
{cl_ros_timer::ClRosTimer ||+ ClRosTimer()
+ ~ClRosTimer()
+ initialize()
+ onTimerTick()
+ onOrthogonalAllocation()
# timerCallback()
}
-onTimerTick_
-postCountDownEvent_
{smacc::SmaccClientBehavior ||+ onEntry()
+ onExit()
}
-timerClient_
{cl_ros_timer::CbTimerCountdownLoop ||+ CbTimerCountdownLoop()
+ onEntry()
+ onExit()
+ onOrthogonalAllocation()
+ onTimerTick()
- onClientTimerTickCallback()
}