

threeThreads

Tight Rope v0.88

4th March 2017

1 ID Files

1.1 MissionIds

section *MissionIds* **parents** *scj_prelude*, *MissionId*

MissionAMID : *MissionID*

distinct $\langle \text{nullMissionId}, \text{MissionAMID} \rangle$

1.2 SchedulablesIds

section *SchedulableIds* **parents** *scj_prelude, SchedulableId*

mainSequencerSID : SchedulableID

MT1SID : SchedulableID

MT2SID : SchedulableID

MT3SID : SchedulableID

distinct(*nullSequencerId, nullSchedulableId, mainSequencerSID,*
MT1SID, MT2SID,
MT3SID)

1.3 Non-Paradigm Objects

1.4 ThreadIds

section *ThreadId* **parents** *scj_prelude, GlobalTypes*

SafeletTid : *ThreadID*
nullThreadId : *ThreadID*

distinct(*SafeletTid*, *nullThreadId*)

1.5 ObjectIds

section *ObjectIds* **parents** *scj_prelude, GlobalTypes*

$distinct \langle \rangle$

2 Network

2.1 Network Channel Sets

section *NetworkChannels* **parents** *scj_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, TopLevelMissionSequencerFWChan, FrameworkChan, SafeletChan, AperiodicEventHandlerChan, ManagedThreadChan, OneShotEventHandlerChan, PeriodicEventHandlerChan, MissionSequencerMethChan*

channelset *TerminateSync* ==
{*schedulables_terminated, schedulables_stopped, get_activeSchedulables*}

channelset *ControlTierSync* ==
{*start_toplevel_sequencer, done_toplevel_sequencer, done_safeletFW*}

channelset *TierSync* ==
{*start_mission . MissionA, done_mission . MissionA, done_safeletFW, done_toplevel_sequencer*}

channelset *MissionSync* ==
{*done_safeletFW, done_toplevel_sequencer, register, signalTerminationCall, signalTerminationRet, activate_schedulables, done_schedulable, cleanupSchedulableCall, cleanupSchedulableRet*}

channelset *SchedulablesSync* ==
{*activate_schedulables, done_safeletFW, done_toplevel_sequencer*}

channelset *ClusterSync* ==
{*done_toplevel_sequencer, done_safeletFW*}

channelset *SafeltAppSync* $\hat{=}$
{*getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end_safelet_app*}

channelset *MissionSequencerAppSync* ==
{*getNextMissionCall, getNextMissionRet, end_sequencer_app*}

channelset *MissionAppSync* ==
{*initializeCall, register, initializeRet, cleanupMissionCall, cleanupMissionRet*}

channelset *AppSync* ==
{*SafeltAppSync, MissionSequencerAppSync, MissionAppSync, MTAppSync, OSEHSync, APEHSync, PEHSync, getSequencer, end_mission_app, end_managedThread_app, setCeilingPriority, requestTerminationCall, requestTerminationRet, terminationPendingCall, terminationPendingRet, handleAsyncEventCall, handleAsyncEventRet*}

channelset *ThreadSync* ==
{*raise_thread_priority, lower_thread_priority, isInterruptedCall, isInterruptedRet, get_priorityLevel*}

channelset *LockingSync* ==
{*lockAcquired, startSyncMeth, endSyncMeth, waitCall, waitRet, notify, isInterruptedCall, isInterruptedRet, interruptedCall, interruptedRet, done_toplevel_sequencer, get_priorityLevel*}

2.2 Locking

section *NetworkLocking* **parents** *scj_prelude, GlobalTypes, FrameworkChan, MissionId, MissionIds, ThreadIds, NetworkChannels, ObjectFW, ThreadFW, Priority*

process *Threads* $\hat{=}$
(**Skip**)

process *Objects* $\hat{=}$
(**Skip**)

process *Locking* $\hat{=}$ (*Threads* \llbracket *ThreadSync* \rrbracket *Objects*) \triangle (*done_toplevel_sequencer* \longrightarrow **Skip**)

2.3 Program

section *Program* **parents** *scj_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MissionFW, SafeletFW, TopLevelMissionSequencerFW, NetworkChannels, ManagedThreadFW, SchedulableMissionSequencerFW, PeriodicEventHandlerFW, OneShotEventHandlerFW, AperiodicEventHandlerFW, ObjectFW, ThreadFW, MyAppApp, mainSequencerApp, MissionAApp, MT1App, MT2App, MT3App*

process *ControlTier* $\hat{=}$

$$\left(\begin{array}{c} \text{SafeletFW} \\ \llbracket \text{ControlTierSync} \rrbracket \\ \text{TopLevelMissionSequencerFW}(\text{mainSequencer}) \end{array} \right)$$

process *Tier0* $\hat{=}$

$$\left(\begin{array}{c} \text{MissionFW}(\text{MissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left(\begin{array}{c} \text{ManagedThreadFW}(\text{MT1ID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{ManagedThreadFW}(\text{MT2ID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{ManagedThreadFW}(\text{MT3ID}) \end{array} \right) \end{array} \right)$$

process *Framework* $\hat{=}$

$$\left(\begin{array}{c} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ (\text{Tier0}) \end{array} \right)$$

process *Application* $\hat{=}$

$$\left(\begin{array}{c} \text{MyAppApp} \\ ||| \\ \text{mainSequencerApp} \\ ||| \\ \text{MissionAApp} \\ ||| \\ \text{MT1App} \\ ||| \\ \text{MT2App} \\ ||| \\ \text{MT3App} \end{array} \right)$$

process *Program* $\hat{=}$ $(\text{Framework} \llbracket \text{AppSync} \rrbracket \text{Application}) \llbracket \text{LockingSync} \rrbracket \text{Locking}$

3 Safelet

section *MyAppApp* **parents** *scj_prelude, SchedulableId, SchedulableIds, SafeletChan, MethodCallBindingChannels*

process *MyAppApp* $\hat{=}$ **begin**

InitializeApplication $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeApplicationCall} \longrightarrow \\ \textit{initializeApplicationRet} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

GetSequencer $\hat{=}$
 $\left(\begin{array}{l} \textit{getSequencerCall} \longrightarrow \\ \textit{getSequencerRet} \text{ ! } \textit{mainSequencerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

Methods $\hat{=}$
 $\left(\begin{array}{l} \textit{GetSequencer} \\ \square \\ \textit{InitializeApplication} \end{array} \right); \textit{Methods}$

• $(\textit{Methods}) \triangle (\textit{end_safelet_app} \longrightarrow \mathbf{Skip})$

end

4 Top Level Mission Sequencer

section *mainSequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *mainSequencerClass*, *MethodCallBindingChannels*

process *mainSequencerApp* $\hat{=}$ **begin**

<i>State</i> <i>this</i> : ref <i>mainSequencerClass</i>

state *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = new <i>mainSequencerClass</i> ()

GetNextMission $\hat{=}$ **var** *ret* : *MissionID* •
 $\left(\begin{array}{l} \text{getNextMissionCall} . \text{mainSequencerSID} \longrightarrow \\ \text{ret} := \text{this} . \text{getNextMission}(); \\ \text{getNextMissionRet} . \text{mainSequencerSID} ! \text{ret} \longrightarrow \\ \text{Skip} \end{array} \right)$

Methods $\hat{=}$
 $(\text{GetNextMission}) ; \text{Methods}$

• $(\text{Init} ; \text{Methods}) \triangle (\text{end_sequencer_app} . \text{mainSequencerSID} \longrightarrow \text{Skip})$

end

section *mainSequencerClass* **parents** *scj_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

class *mainSequencerClass* $\hat{=}$ **begin**

state <i>State</i> <i>notReleased</i> : \mathbb{B}

state *State*

initial <i>Init</i> <i>State</i> '
<i>notReleased</i> ' = <i>true</i>

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } notReleased \longrightarrow \\ \quad \left(notReleased := \mathbf{False}; \right. \\ \quad \left. ret := MissionAMID \right) \\ \parallel \neg notReleased \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5 Missions

5.1 MissionA

section *MissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
 SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MissionAMethChan*
 , *MethodCallBindingChannels*

process *MissionApp* $\hat{=}$ **begin**

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MissionAMID} \longrightarrow \\ \textit{register} ! \textit{MT1SID} ! \textit{MissionAMID} \longrightarrow \\ \textit{register} ! \textit{MT2SID} ! \textit{MissionAMID} \longrightarrow \\ \textit{register} ! \textit{MT3SID} ! \textit{MissionAMID} \longrightarrow \\ \textit{initializeRet} . \textit{MissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \textit{cleanupMissionCall} . \textit{MissionAMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MissionAMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

Methods $\hat{=}$ $\left(\begin{array}{c} \textit{InitializePhase} \\ \square \\ \textit{CleanupPhase} \end{array} \right) ; \textit{Methods}$

$\bullet (\textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MissionAMID} \longrightarrow \mathbf{Skip})$

end

5.2 Schedulables of MissionA

section *MT1App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

process *MT1App* $\hat{=}$ **begin**

Run $\hat{=}$
$$\left(\begin{array}{l} \text{runCall} . \text{MT1SID} \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . \text{MT1SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

Methods $\hat{=}$
 $(\text{Run}) ; \text{Methods}$

• $(\text{Methods}) \triangle (\text{end_managedThread_app} . \text{MT1SID} \longrightarrow \mathbf{Skip})$

end

section *MT2App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

process *MT2App* $\hat{=}$ **begin**

Run $\hat{=}$

$$\left(\begin{array}{l} \text{runCall} . \text{MT2SID} \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . \text{MT2SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

Methods $\hat{=}$
 $(\text{Run}) ; \text{Methods}$

• $(\text{Methods}) \triangle (\text{end_managedThread_app} . \text{MT2SID} \longrightarrow \mathbf{Skip})$

end

section *MT3App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

process *MT3App* $\hat{=}$ **begin**

Run $\hat{=}$

$$\left(\begin{array}{l} \text{runCall} . \text{MT3SID} \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . \text{MT3SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

Methods $\hat{=}$
 $(\text{Run}) ; \text{Methods}$

$\bullet (\text{Methods}) \triangle (\text{end_managedThread_app} . \text{MT3SID} \longrightarrow \mathbf{Skip})$

end