

ThreeThreadsExample

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

<i>MissionAID</i> : <i>MissionID</i>
<i>distinct</i> (<i>nullMissionId</i> , <i>MissionAID</i>)

1.2 SchedulablesIds

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

mainSequencerID : *SchedulableID*

MT1ID : *SchedulableID*

MT2ID : *SchedulableID*

MT3ID : *SchedulableID*

distinct \langle *nullSequencerId*, *nullSchedulableId*, *mainSequencerIDID*,
MT1ID, *MT2ID*,
MT3ID \rangle

1.3 ThreadIds

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

MT2ThreadId : *ThreadId*

MT1ThreadId : *ThreadId*

MT3ThreadId : *ThreadId*

distinct (*SafeletThreadId*, *nullThreadId*,
MT2ThreadId, *MT1ThreadId*,
MT3ThreadId)

1.4 ObjectIds

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

MyAppObjectID : *ObjectID*
MissionAObjectID : *ObjectID*
MT1ObjectID : *ObjectID*
MT2ObjectID : *ObjectID*
MT3ObjectID : *ObjectID*

distinct (*MyAppObjectID*, *MissionAObjectID*,
MT1ObjectID, *MT2ObjectID*,
MT3ObjectID)

2 Network

2.1 Network Channel Sets

section *NetworkChannels* **parents** *scj_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableChan, TopLevelMissionSequencerFWChan, FrameworkChan, SafeletChan*

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 *AppSync* ==
 { *SafeltAppSync, MissionSequencerAppSync, MissionAppSync, MTAppSync, OSEHSync, APEHSync, 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 MethodCallBinder

channelset *MethodCallBinderSync* == { *done_toplevel_sequencer*, }

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

BinderActions $\hat{=}$
) (

- *BinderActions* \triangle (*done_toplevel_sequencer* \longrightarrow **Skip**)

end

process *ApplicationB* $\hat{=}$ *Application* [*MethodCallBinderSync*] *MethodCallBinder*

2.3 Locking

process *Threads* $\hat{=}$
 $\left(\begin{array}{l} \textit{ThreadFW}(\textit{MT2ThreadID},) \\ ||| \\ \textit{ThreadFW}(\textit{MT1ThreadID},) \\ ||| \\ \textit{ThreadFW}(\textit{MT3ThreadID},) \end{array} \right)$

process *Objects* $\hat{=}$
 $\left(\begin{array}{l} \textit{ObjectFW}(\textit{MyAppObjectID}) \\ ||| \\ \textit{ObjectFW}(\textit{MissionAObjectID}) \\ ||| \\ \textit{ObjectFW}(\textit{MT1ObjectID}) \\ ||| \\ \textit{ObjectFW}(\textit{MT2ObjectID}) \\ ||| \\ \textit{ObjectFW}(\textit{MT3ObjectID}) \end{array} \right)$

process *Locking* $\hat{=}$ *Threads* \llbracket *ThreadSync* \rrbracket *Objects*

2.4 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* $\hat{=}$

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

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

3 Safelet

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

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

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

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

immortalMemorySizeMeth $\hat{=}$ **var** *ret* : \mathbb{Z} •
 $\left(\begin{array}{l} \textit{immortalMemorySizeCall} . \textit{MyApp} \longrightarrow \\ (\textit{ret} := \textit{Const.IMMORTAL_MEM_DEFAULT}) ; \\ \textit{immortalMemorySizeRet} . \textit{MyApp} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

• (*Methods*) \triangle (*end_safelet_app* \longrightarrow **Skip**)

end

4 Top Level Mission Sequencer

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

process *mainSequencerApp* $\hat{=}$
name : *String* • **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* •
 $\left(\begin{array}{l} \textit{getNextMissionCall} . \textit{mainSequencer} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{mainSequencer} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

• $(\textit{Init} ; \textit{Methods}) \triangle (\textit{end_sequencer_app} . \textit{mainSequencer} \longrightarrow \mathbf{Skip})$

end

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* $\hat{=}$ **var** *ret* : *MissionID* •
$$\left(\begin{array}{l} \text{if } \textit{notReleased} = \textbf{True} \longrightarrow \\ \quad \left(\textbf{var } \textit{mission} : \textit{MissionID} \bullet \textit{mission} := \textit{MissionA}; \right. \\ \quad \left. \textit{ret} := \textit{mission} \right) \\ \parallel \textit{notReleased} = \textbf{True} \longrightarrow \\ \quad (\textit{ret} := \textit{nullMissionId}) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5 Missions

5.1 MissionA

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

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

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

state *State*

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

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MissionA} \longrightarrow \\ \textit{register} ! \textit{MT1} ! \textit{MissionA} \longrightarrow \\ \textit{register} ! \textit{MT2} ! \textit{MissionA} \longrightarrow \\ \textit{register} ! \textit{MT3} ! \textit{MissionA} \longrightarrow \\ \textit{initializeRet} . \textit{MissionA} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

• (*Init* ; *Methods*) \triangle (*end_mission_app* . *MissionA* \longrightarrow **Skip**)

end

5.2 Schedulables of MissionA

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

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

Run $\hat{=}$
 $\left(\begin{array}{l} \textit{runCall} . \textit{MT1} \longrightarrow \\ (\mathbf{Skip}) ; \\ \textit{runRet} . \textit{MT1} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Methods}) \triangle (\textit{end_managedThread_app} . \textit{MT1} \longrightarrow \mathbf{Skip})$

end

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

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

Run $\hat{=}$

$$\left(\begin{array}{l} \text{runCall} . MT2 \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . MT2 \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

• $(Methods) \triangle (end_managedThread_app . MT2 \longrightarrow \mathbf{Skip})$

end

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

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

Run $\hat{=}$

$$\left(\begin{array}{l} \text{runCall} . MT3 \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . MT3 \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

$\bullet (Methods) \triangle (end_managedThread_app . MT3 \longrightarrow \mathbf{Skip})$

end