Mission+PEH+APEH(mission1)

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

$$\begin{array}{|l}
MyMissionID : MissionID \\
\hline
distinct \langle nullMissionId, MyMissionID \rangle
\end{array}$$

1.2 SchedulablesIds

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

$$
\begin{array}{|l}
mainSequencerID : SchedulableID \\
APEHID : SchedulableID \\
PEHID : SchedulableID \\
\hline
distinct \langle nullSequencerId, nullSchedulableId, mainSequencerIDID, \\
APEHID, PEHID \rangle
\end{array}
$$

1.3 ThreadIds

section *ThreadIds* **parents** *scj_prelude*, *GlobalTypes*

$$
\begin{array}{|l}
PEHThreadID : ThreadID \\
APEHThreadID : ThreadID \\
\hline
distinct\langle SafeletThreadId, nullThreadId, \\
PEHThreadID, APEHThreadID\rangle
\end{array}
$$

1.4 ObjectIds

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

$$
\begin{array}{|l}
MyAppObjectID : ObjectID \\
MyMissionObjectID : ObjectID \\
APEHObjectID : ObjectID \\
PEHObjectID : ObjectID \\
\hline
distinct\langle MyAppObjectID, MyMissionObjectID, \\
APEHObjectID, PEHObjectID\rangle
\end{array}
$$

2 Network

2.1 Network Channel Sets

section $\mathit{NetworkChannels}$ **parents** $\mathit{scj\_prelude}, \mathit{MissionId}, \mathit{MissionIds},$
$\mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{MissionChan}, \mathit{SchedulableChan}, \mathit{TopLevelMissionSequencerFWChan},$
$\mathit{FrameworkChan}, \mathit{SafeletChan}$

channelset $\mathit{TerminateSync} ==$
$\{\!| \mathit{schedulables\_terminated}, \mathit{schedulables\_stopped}, \mathit{get\_activeSchedulables} |\!\}$

channelset $\mathit{ControlTierSync} ==$
$\{\!| \mathit{start\_toplevel\_sequencer}, \mathit{done\_toplevel\_sequencer}, \mathit{done\_safeletFW} |\!\}$

channelset $\mathit{TierSync} ==$
$\{\!| \mathit{start\_mission} . \mathit{MyMission}, \mathit{done\_mission} . \mathit{MyMission},$
$\mathit{done\_safeletFW}, \mathit{done\_toplevel\_sequencer} |\!\}$

channelset $\mathit{MissionSync} ==$
$\{\!| \mathit{done\_safeletFW}, \mathit{done\_toplevel\_sequencer}, \mathit{register},$
$\mathit{signalTerminationCall}, \mathit{signalTerminationRet}, \mathit{activate\_schedulables}, \mathit{done\_schedulable},$
$\mathit{cleanupSchedulableCall}, \mathit{cleanupSchedulableRet} |\!\}$

channelset $\mathit{SchedulablesSync} ==$
$\{\!| \mathit{activate\_schedulables}, \mathit{done\_safeletFW}, \mathit{done\_toplevel\_sequencer} |\!\}$

channelset $\mathit{ClusterSync} ==$
$\{\!| \mathit{done\_toplevel\_sequencer}, \mathit{done\_safeletFW} |\!\}$

channelset $\mathit{AppSync} ==$
$\bigcup\{\mathit{SafeltAppSync}, \mathit{MissionSequencerAppSync}, \mathit{MissionAppSync},$
$\mathit{MTAppSync}, \mathit{OSEHSync}, \mathit{APEHSync},$
$\{\!| \mathit{getSequencer}, \mathit{end\_mission\_app}, \mathit{end\_managedThread\_app},$
$\mathit{setCeilingPriority}, \mathit{requestTerminationCall}, \mathit{requestTerminationRet}, \mathit{terminationPendingCall},$
$\mathit{terminationPendingRet}, \mathit{handleAsyncEventCall}, \mathit{handleAsyncEventRet} |\!\}\}$

channelset $\mathit{ThreadSync} ==$
$\{\!| \mathit{raise\_thread\_priority}, \mathit{lower\_thread\_priority}, \mathit{isInterruptedCall}, \mathit{isInterruptedRet}, \mathit{get\_priorityLevel} |\!\}$

channelset $\mathit{LockingSync} ==$
$\{\!| \mathit{lockAcquired}, \mathit{startSyncMeth}, \mathit{endSyncMeth}, \mathit{waitCall}, \mathit{waitRet}, \mathit{notify}, \mathit{isInterruptedCall}, \mathit{isInterruptedRet},$
$\mathit{interruptedCall}, \mathit{interruptedRet}, \mathit{done\_toplevel\_sequencer}, \mathit{get\_priorityLevel} |\!\}$

2.2 MethodCallBinder

channelset $MethodCallBinderSync == \lbrace\!| done\_toplevel\_sequencer, |\!\rbrace$

process $MethodCallBinder \mathrel{\widehat=}$ **begin**

$BinderActions \mathrel{\widehat=}$
$)($

$\bullet\ BinderActions \mathbin{\triangle} (done\_toplevel\_sequencer \longrightarrow \mathbf{Skip})$

end

process $ApplicationB \mathrel{\widehat=} Application \llbracket MethodCallBinderSync \rrbracket MethodCallBinder$

2.3 Locking

process *Threads* $\widehat{=}$

$$\left(\begin{array}{l} ThreadFW(PEHThreadID,) \\ ||| \\ ThreadFW(APEHThreadID,) \end{array}\right)$$

process *Objects* $\widehat{=}$

$$\left(\begin{array}{l} ObjectFW(MyAppObjectID) \\ ||| \\ ObjectFW(MyMissionObjectID) \\ ||| \\ ObjectFW(APEHObjectID) \\ ||| \\ ObjectFW(PEHObjectID) \end{array}\right)$$

process *Locking* $\widehat{=}$ *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*, *MyMissionApp*, *APEHApp*, *PEHApp*

process *ControlTier* $\widehat{=}$

$$\left(\begin{array}{l} SafeletFW \\ \quad [\![ControlTierSync]\!] \\ TopLevelMissionSequencerFW(mainSequencer) \end{array}\right)$$

process *Tier0* $\widehat{=}$

$$\left(\begin{array}{l} MissionFW(MyMissionID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} AperiodicEventHandlerFW(APEHID) \\ \quad [\![SchedulablesSync]\!] \\ PeriodicEventHandlerFW(PEHID) \end{array}\right) \end{array}\right)$$

process *Framework* $\widehat{=}$

$$\left(\begin{array}{l} ControlTier \\ \quad [\![TierSync]\!] \\ (Tier0) \end{array}\right)$$

process *Application* $\widehat{=}$

$$\left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MyMissionApp \\ ||| \\ APEHApp(AapParams, MyMissionID) \\ ||| \\ PEHApp(ApParams, apehID) \end{array}\right)$$

process *Program* $\widehat{=}$ $(Framework \; [\![\; AppSync \;]\!] \; ApplicationB) \; [\![\; LockingSync \;]\!] \; Locking$

3 Safelet

$\mathbf{section}\ \mathit{MyAppApp}\ \mathbf{parents}\ \mathit{scj\_prelude}, \mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{SafeletChan}$

$\mathbf{process}\ \mathit{MyAppApp} \mathrel{\widehat=} \mathbf{begin}$

$$\mathit{InitializeApplication} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{initializeApplicationCall} \longrightarrow \\ \mathit{initializeApplicationRet} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{GetSequencer} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{getSequencerCall} \longrightarrow \\ \mathit{getSequencerRet}\,!\,\mathit{mainSequencerID} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{immortalMemorySizeMeth} \mathrel{\widehat=} \mathbf{var}\ \mathit{ret} : \mathbb{Z} \bullet \left(\begin{array}{l} \mathit{immortalMemorySizeCall}\,.\,\mathit{MyApp} \longrightarrow \\ \left(\mathit{ret} := \mathit{Const.IMMORTAL\_MEM\_DEFAULT}\right); \\ \mathit{immortalMemorySizeRet}\,.\,\mathit{MyApp}\,!\,\mathit{ret} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{Methods} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{GetSequencer} \\ \Box \\ \mathit{InitializeApplication} \\ \Box \\ \mathit{immortalMemorySizeMeth} \end{array}\right);\ \mathit{Methods}$$

$\bullet\ (\mathit{Methods}) \bigtriangleup (\mathit{end\_safelet\_app} \longrightarrow \mathbf{Skip})$

$\mathbf{end}$

4 Top Level Mission Sequencer

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

process *mainSequencerApp* $\widehat{=}$
name : *String* ● **begin**

State

$this : \mathbf{ref}\ mainSequencerClass$

state *State*

Init

$State\,'$

$this' = \mathbf{new}\ mainSequencerClass()$

$$GetNextMission \mathrel{\widehat{=}} \mathbf{var}\ ret : MissionID \bullet \left(\begin{array}{l} getNextMissionCall\,.\,mainSequencer \longrightarrow \\ ret := this\,.\,getNextMission(); \\ getNextMissionRet\,.\,mainSequencer\,!\,ret \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$Methods \mathrel{\widehat{=}} \left(GetNextMission\right)\ ;\ Methods$$

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_sequencer\_app\,.\,mainSequencer \longrightarrow \mathbf{Skip})$$

end

class $\mathit{mainSequencerClass} \mathrel{\widehat=}$ **begin**

$$
\begin{array}{|l}
\textbf{state}\ \mathit{State} \\
\hline
\mathit{notReleased} : \mathbb{B} \\
\hline
\end{array}
$$

state State

$$
\begin{array}{|l}
\textbf{initial}\ \mathit{Init} \\
\hline
\mathit{State}\,' \\
\hline
\mathit{notReleased}' = \mathit{true} \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textbf{protected}\ \mathit{getNextMission} \mathrel{\widehat=} \textbf{var}\ \mathit{ret} : \mathit{MissionID} \bullet \\
\left(\begin{array}{l}
\textbf{if}\ \ \mathit{notReleased} = \textbf{True} \longrightarrow \\
\qquad \left(\begin{array}{l}
\textbf{var}\ \mathit{mission} : \mathit{MissionID} \bullet \mathit{mission} := \mathit{MyMission}\ ; \\
\mathit{this}\ .\ \mathit{notReleased} := \mathit{false}; \\
\mathit{ret} := \mathit{mission}
\end{array}\right) \\
[\!]\ \mathit{notReleased} = \textbf{True} \longrightarrow \\
\qquad \left(\mathit{ret} := \mathit{nullMissionId}\right) \\
\textbf{fi}
\end{array}\right)
\end{array}
$$

$\bullet$ **Skip**

end

5 Missions

5.1 MyMission

section $\mathit{MyMissionApp}$ **parents** $\mathit{scj\_prelude}, \mathit{MissionId}, \mathit{MissionIds},$
$\mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{MissionChan}, \mathit{SchedulableMethChan}$
$, \mathit{MyMissionMethChan}$

process $\mathit{MyMissionApp} \mathrel{\widehat=}$ **begin**

$$\begin{array}{|l}
\hline \mathit{State} \\
\mathit{this} : \mathbf{ref}\ \mathit{MyMissionClass} \\
\hline
\end{array}$$

state State

$$\begin{array}{|l}
\hline \mathit{Init} \\
\mathit{State}\,' \\
\hline
\mathit{this}' = \mathbf{new}\ \mathit{MyMissionClass}() \\
\hline
\end{array}$$

$$\mathit{InitializePhase} \mathrel{\widehat=} \\ \left(\begin{array}{l} \mathit{initializeCall}\,.\,\mathit{MyMission} \longrightarrow \\ \mathit{register}\,!\,\mathit{APEH}\,!\,\mathit{MyMission} \longrightarrow \\ \mathit{register}\,!\,\mathit{PEH}\,!\,\mathit{MyMission} \longrightarrow \\ \mathit{initializeRet}\,.\,\mathit{MyMission} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{CleanupPhase} \mathrel{\widehat=} \\ \left(\begin{array}{l} \mathit{cleanupMissionCall}\,.\,\mathit{MyMission} \longrightarrow \\ \mathit{cleanupMissionRet}\,.\,\mathit{MyMission}\,!\,\mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{Methods} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{InitializePhase} \\ \Box \\ \mathit{CleanupPhase} \end{array}\right) \; ; \; \mathit{Methods}$$

$$\bullet\ (\mathit{Init}\ ;\ \mathit{Methods}) \mathbin{\triangle} (\mathit{end\_mission\_app}\,.\,\mathit{MyMission} \longrightarrow \mathbf{Skip})$$

end

5.2 Schedulables of MyMission

section *APEHApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*

process $APEHApp \mathrel{\widehat=}$
$\quad controllingMission : MissionID \bullet$ **begin**

$handleAsyncEvent \mathrel{\widehat=}$

$$\left(\begin{array}{l} handleAsyncEventCall \,.\, APEH \longrightarrow \\ \big(requestTerminationCall \,.\, controllingMission \,.\, APEH \longrightarrow \quad requestTerminationRet \,.\, controllingMission \,.\, APEH \,?\, reques \\ handleAsyncEventRet \,.\, APEH \longrightarrow \\ \mathbf{Skip} \end{array}\right.$$

$Methods \mathrel{\widehat=}$
$\big(handleAsyncEvent\big) \; ; \;\; Methods$

$\bullet\ (Methods) \mathrel{\triangle} (end\_aperiodic\_app \,.\, APEH \longrightarrow \mathbf{Skip})$

end

class $APEHClass \mathrel{\widehat=}$ **begin**

state $State$

$controllingMission : Mission$

state $State$

initial $Init$

$State\,'$

$\bullet$ **Skip**

end

section *PEHApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*

process *PEHApp* $\widehat{=}$
    $apeh : SchedulableID \bullet$ **begin**

$handleAsyncEvent \mathrel{\widehat{=}}$
$$\left(\begin{array}{l} handleAsyncEventCall \,.\, PEH \longrightarrow \\ \big(releaseCall \,.\, apeh \,.\, PEH \longrightarrow \quad releaseRet \,.\, apeh \,.\, PEH \,?\, release \longrightarrow \quad \mathbf{Skip}\big) \,; \\ handleAsyncEventRet \,.\, PEH \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$Methods \mathrel{\widehat{=}}$
$\big(handleAsyncEvent\big) \;;\; Methods$

$\bullet\ (Methods) \mathrel{\triangle} (end\_periodic\_app \,.\, PEH \longrightarrow \mathbf{Skip})$

end

class $PEHClass \mathrel{\widehat=}$ **begin**

state $State$

$apeh : AperiodicEventHandler$

state $State$

initial $Init$

$State\,'$

$\bullet$ **Skip**

end