mission1

Tight Rope v0.75

13th February 2017

1 ID Files

1.1 MissionIds

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

$$MyMissionMID : MissionID$$

$$distinct\langle nullMissionId, MyMissionMID\rangle$$

1.2 SchedulablesIds

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

$$
\begin{array}{|l}
mainSequencerSID : SchedulableID \\
APEHSID : SchedulableID \\
PEHSID : SchedulableID \\
\hline
distinct\langle nullSequencerId, nullSchedulableId, mainSequencerSID, \\
APEHSID, PEHSID\rangle
\end{array}
$$

1.3 Non-Paradigm Objects

1.4 ThreadIds

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

$$SafeletTId : ThreadID$$
$$nullThreadId : ThreadID$$

$$distinct\langle SafeletTId, nullThreadId\rangle$$

1.5 ObjectIds

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

$$\vdash \; distinct\langle\rangle$$

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* . *MyMission*, *done_mission* . *MyMission*,
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* $\widehat{=}$
{| *getSequencerCall*, *getSequencerRet*, *initializeApplicationCall*, *initializeApplicationRet*, *end_safelet_app* |}

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

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

channelset *AppSync* ==
$\bigcup\{$*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* $\widehat{=}$
$(\mathbf{Skip})$

process *Objects* $\widehat{=}$
$(\mathbf{Skip})$

process *Locking* $\widehat{=}$ *Threads* $[\![$ *ThreadSync* $]\!]$ *Objects*

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*, *MyMissionApp*, *APEHApp*, *PEHApp*

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

process $Tier0 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(MyMissionID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} AperiodicEventHandlerFW(APEHID, aperiodic, (time(5,0), nullSchedulableId)) \\ \quad [\![SchedulablesSync]\!] \\ PeriodicEventHandlerFW(PEHID, (time(60,0), time(5,0), NULL, nullSchedulableId)) \end{array}\right) \end{array}\right)$$

process $Framework \mathrel{\widehat=}$
$$\left(\begin{array}{l} ControlTier \\ \quad [\![TierSync]\!] \\ (\, Tier0) \end{array}\right)$$

process $Application \mathrel{\widehat=}$
$$\left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MyMissionApp \\ ||| \\ APEHApp(MyMissionID) \\ ||| \\ PEHApp(apehID) \end{array}\right)$$

process $Program \mathrel{\widehat=} \big(Framework \;[\![\; AppSync \;]\!]\; Application\big) \;[\![\; LockingSync \;]\!]\; Locking$

3 Safelet

section $MyAppApp$ **parents** $scj\_prelude, SchedulableId, SchedulableIds, SafeletChan, MethodCallBindingChannels$

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

$$InitializeApplication \mathrel{\widehat=} \begin{pmatrix} initializeApplicationCall \longrightarrow \\ initializeApplicationRet \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$GetSequencer \mathrel{\widehat=} \begin{pmatrix} getSequencerCall \longrightarrow \\ getSequencerRet\,!\,mainSequencerSID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} GetSequencer \\ \Box \\ InitializeApplication \end{pmatrix} ;\ Methods$$

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

end

4 Top Level Mission Sequencer

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

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

State

$this : \mathbf{ref}\ mainSequencerClass$

state *State*

Init

$State'$

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

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

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

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

end

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

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

protected *getNextMission* $\widehat{=}$ **var** *ret* : *MissionID* •

$$\left(\begin{array}{l} \textbf{Skip}; \\ \textbf{if } notReleased = \textbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} this\,.\,notReleased := \textbf{False}; \\ ret := MyMissionMID \end{array}\right) \\ [\!] \neg\, notReleased = \textbf{True} \longrightarrow \\ \quad \left(ret := nullMissionId\right) \\ \textbf{fi} \end{array}\right)$$

• **Skip**

end

5 Missions

5.1 MyMission

section *MyMissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MyMissionMethChan*, *MethodCallBindingChannels*

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

State

$this : \mathbf{ref}\ MyMissionClass$

state *State*

Init

$State'$

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,MyMissionMID \longrightarrow \\ register\,!\,APEHSID\,!\,MyMissionMID \longrightarrow \\ register\,!\,PEHSID\,!\,MyMissionMID \longrightarrow \\ initializeRet\,.\,MyMissionMID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \mathbf{var}\,\mathbb{B} : ret \bullet cleanupMissionCall\,.\,MyMissionMID \longrightarrow \\ cleanupMissionRet\,.\,MyMissionMID\,!\,\mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

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

end

5.2 Schedulables of MyMission

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

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

$$handleAsyncEvent \mathrel{\widehat=}$$
$$\left(\begin{array}{l} handleAsyncEventCall\,.\,APEHSID \longrightarrow \\ \left(\begin{array}{l} \textbf{Skip}; \\ requestTerminationCall\,.\,controllingMission\,.\,APEHSID \longrightarrow \\ requestTerminationRet\,.\,controllingMission\,.\,APEHSID\,?\,requestTermination \longrightarrow \\ \textbf{Skip} \end{array}\right); \\ handleAsyncEventRet\,.\,APEHSID \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

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

$$\bullet\ (Methods) \triangle (end\_aperiodic\_app\,.\,APEHSID \longrightarrow \textbf{Skip})$$

end

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

process $PEHApp \mathrel{\widehat=}$
$\quad apeh : SchedulableID \bullet \textbf{begin}$

$$handleAsyncEvent \mathrel{\widehat=} \left(\begin{array}{l} handleAsyncEventCall\,.\,PEHSID \longrightarrow \\ \left(\begin{array}{l}\textbf{Skip};\\ release\,.\,apeh \longrightarrow\end{array}\right);\\ \textbf{Skip} \\ handleAsyncEventRet\,.\,PEHSID \longrightarrow \\ \textbf{Skip}\end{array}\right)$$

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

$$\bullet\ (Methods) \triangle (end\_periodic\_app\,.\,PEHSID \longrightarrow \textbf{Skip})$$

end