Mission+MT+OSEH(mission2)

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

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

1.2 SchedulablesIds

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

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

1.3 ThreadIds

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

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

1.4 ObjectIds

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

$$
\begin{array}{|l}
MyAppObjectID : ObjectID \\
MissionAObjectID : ObjectID \\
OSEHObjectID : ObjectID \\
MTObjectID : ObjectID \\
\hline
distinct \langle MyAppObjectID, MissionAObjectID, \\
OSEHObjectID, MTObjectID \rangle
\end{array}
$$

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* ==
$\bigcup\{$ *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

$\mathbf{channel}\ binder\_systemActionCall : MissionID \times SchedulableID$
$\mathbf{channel}\ binder\_systemActionRet : MissionID \times SchedulableID$

$systemActionLocs == \{MissionA\}$
$systemActionCallers == \{MT\}$

$\mathbf{channelset}\ MethodCallBinderSync == \{\!|\ done\_toplevel\_sequencer, binder\_systemActionCall, binder\_systemActionRet\ |\!\}$

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

$$systemAction\_MethodBinder \mathrel{\widehat=} \left(\begin{array}{l} binder\_systemActionCall \\ \quad ?\,loc : (loc \in systemActionLocs) \\ \quad ?\,caller : (caller \in systemActionCallers) \longrightarrow \\ systemActionCall\,.\,loc\,.\,caller \longrightarrow \\ systemActionRet\,.\,loc\,.\,caller \longrightarrow \\ binder\_systemActionRet\,.\,loc\,.\,caller \longrightarrow \\ systemAction\_MethodBinder \end{array}\right)$$

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

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

$\mathbf{end}$

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

2.3 Locking

$$\mathbf{process}\ \mathit{Threads} \mathrel{\widehat{=}} \left(\begin{array}{l} \mathit{ThreadFW}(\mathit{OSEHThreadID},) \\ ||| \\ \mathit{ThreadFW}(\mathit{MTThreadID},) \end{array}\right)$$

$$\mathbf{process}\ \mathit{Objects} \mathrel{\widehat{=}} \left(\begin{array}{l} \mathit{ObjectFW}(\mathit{MyAppObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{MissionAObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{OSEHObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{MTObjectID}) \end{array}\right)$$

$$\mathbf{process}\ \mathit{Locking} \mathrel{\widehat{=}} \mathit{Threads} \llbracket \mathit{ThreadSync} \rrbracket \mathit{Objects}$$

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*, *MTApp*, *OSEHApp*

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

$$\textbf{process}\ Tier0 \mathrel{\widehat=} \left(\begin{array}{l} MissionFW(MissionAID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} ManagedThreadFW(MTID) \\ OneShotEventHandlerFW(OSEHID) \\ \quad [\![SchedulablesSync]\!] \end{array}\right) \end{array}\right)$$

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

$$\textbf{process}\ Application \mathrel{\widehat=} \left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MissionAApp \\ ||| \\ MTApp(MissionAID) \\ ||| \\ OSEHApp(RelativeTime, AapParams, MissionAID) \end{array}\right)$$

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

3 Safelet

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

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

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

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

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

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

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

end

4 Top Level Mission Sequencer

section $\mathit{mainSequencerApp}$ **parents** $\mathit{TopLevelMissionSequencerChan}$,
$\mathit{MissionId}, \mathit{MissionIds}, \mathit{SchedulableId}, \mathit{mainSequencerClass}$

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

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

state State

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

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

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

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

end

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State~'$$

$$notReleased' = true$$

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

$\bullet$ **Skip**

end

5 Missions

5.1 MissionA

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

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

$$\begin{array}{|l}
\hline
\textit{State} \\
\quad this : \mathbf{ref}\ MissionAClass \\
\hline
\end{array}$$

state *State*

$$\begin{array}{|l}
\hline
\textit{Init} \\
\quad State' \\
\hline
\quad this' = \mathbf{new}\ MissionAClass() \\
\hline
\end{array}$$

$$InitializePhase \mathrel{\widehat{=}} \left(\begin{array}{l} initializeCall\ .\ MissionA \longrightarrow \\ register\ !\ OSEH\ !\ MissionA \longrightarrow \\ register\ !\ MT\ !\ MissionA \longrightarrow \\ initializeRet\ .\ MissionA \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

$$systemActionMeth \mathrel{\widehat{=}} \left(\begin{array}{l} systemActionCall\ .\ MissionA \longrightarrow \\ this\ .\ systemAction(); \\ systemActionRet\ .\ MissionA \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

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

end

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

public $systemAction \mathrel{\widehat=}$
$(\mathbf{Skip})$

$\bullet$ **Skip**

end

5.2 Schedulables of MissionA

section *MTApp* **parents** *ManagedThreadChan*, *SchedulableId*, *SchedulableIds*
,
MissionAMethChan

process *MTApp* $\widehat{=}$
 controllingMission : *MissionID* • **begin**

$Run \mathrel{\widehat=}$
$\left(\begin{array}{l} runCall\,.\,MT \longrightarrow \\ \big(binder\_systemActionCall\,.\,controllingMission\,.\,MT \longrightarrow \quad binder\_systemActionRet\,.\,controllingMission\,.\,MT \longrightarrow \quad \mathbf{Ski} \\ runRet\,.\,MT \longrightarrow \\ \mathbf{Skip} \end{array}\right.$

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

• $(Methods) \mathbin{\triangle} (end\_managedThread\_app\,.\,MT \longrightarrow \mathbf{Skip})$

end

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

state $State$

$controllingMission : MissionA$

state $State$

initial $Init$

$State\,'$

• **Skip**

end

section *OSEHApp* **parents** *OneShotEventHandlerChan*, *SchedulableId*, *SchedulableIds*

process *OSEHApp* $\widehat{=}$
    $start : HighResolutionTime,$
$controllingMission : MissionID \bullet$ **begin**

$handleAsyncEvent \widehat{=}$
$$\left(\begin{array}{l} handleAsyncEventCall \,.\, OSEH \longrightarrow \\ \left(requestTerminationCall \,.\, controllingMision \,.\, OSEH \longrightarrow \quad requestTerminationRet \,.\, controllingMision \,.\, OSEH \,?\, requestT \right. \\ handleAsyncEventRet \,.\, OSEH \longrightarrow \\ \mathbf{Skip} \end{array}\right.$$

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

$\bullet\ (Methods) \vartriangle (end\_oneShot\_app \,.\, OSEH \longrightarrow \mathbf{Skip})$

end

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

state State

$\mathit{controllingMision} : \mathit{Mission}$

state State

initial Init

State'

$\bullet$ **Skip**

end