TwoSequentialMissions

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

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

1.2 SchedulablesIds

section $\mathit{SchedulableIds}$ **parents** $\mathit{scj\_prelude}, \mathit{SchedulableId}$

$$
\begin{array}{l}
\mathit{mainSequencerID} : \mathit{SchedulableID} \\
\mathit{MT2ID} : \mathit{SchedulableID} \\
\mathit{MT1ID} : \mathit{SchedulableID} \\
\hline
\mathit{distinct} \langle \mathit{nullSequencerId}, \mathit{nullSchedulableId}, \mathit{mainSequencerIDID}, \\
\mathit{MT2ID}, \mathit{MT1ID} \rangle
\end{array}
$$

1.3 ThreadIds

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

$MT2ThreadID : ThreadID$
$MT1ThreadID : ThreadID$

$distinct \langle SafeletThreadId, nullThreadId,$
$MT2ThreadID, MT1ThreadID \rangle$

1.4 ObjectIds

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

$$
\begin{array}{|l}
MyAppObjectID : ObjectID \\
MissionAObjectID : ObjectID \\
MT2ObjectID : ObjectID \\
MissionBObjectID : ObjectID \\
MT1ObjectID : ObjectID \\
\hline
distinct\langle MyAppObjectID, MissionAObjectID, \\
MT2ObjectID, MissionBObjectID, \\
MT1ObjectID\rangle
\end{array}
$$

2 Network

2.1 Network Channel Sets

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

channelset *TerminateSync* ==
$\lbrace\!|$ *schedulables_terminated*, *schedulables_stopped*, *get_activeSchedulables* $|\!\rbrace$

channelset *ControlTierSync* ==
$\lbrace\!|$ *start_toplevel_sequencer*, *done_toplevel_sequencer*, *done_safeletFW* $|\!\rbrace$

channelset *TierSync* ==
$\lbrace\!|$ *start_mission . MissionA*, *done_mission . MissionA*,
done_safeletFW, *done_toplevel_sequencer* $|\!\rbrace$

channelset *TierSync* ==
$\lbrace\!|$ *start_mission . MissionB*, *done_mission . MissionB*,
done_safeletFW, *done_toplevel_sequencer* $|\!\rbrace$

channelset *MissionSync* ==
$\lbrace\!|$ *done_safeletFW*, *done_toplevel_sequencer*, *register*,
signalTerminationCall, *signalTerminationRet*, *activate_schedulables*, *done_schedulable*,
cleanupSchedulableCall, *cleanupSchedulableRet* $|\!\rbrace$

channelset *SchedulablesSync* ==
$\lbrace\!|$ *activate_schedulables*, *done_safeletFW*, *done_toplevel_sequencer* $|\!\rbrace$

channelset *ClusterSync* ==
$\lbrace\!|$ *done_toplevel_sequencer*, *done_safeletFW* $|\!\rbrace$

channelset *AppSync* ==
$\bigcup\{$*SafeltAppSync*, *MissionSequencerAppSync*, *MissionAppSync*,
MTAppSync, *OSEHSync*, *APEHSync*,
$\lbrace\!|$ *getSequencer*, *end_mission_app*, *end_managedThread_app*,
setCeilingPriority, *requestTerminationCall*, *requestTerminationRet*, *terminationPendingCall*,
terminationPendingRet, *handleAsyncEventCall*, *handleAsyncEventRet* $|\!\rbrace\}$

channelset *ThreadSync* ==
$\lbrace\!|$ *raise_thread_priority*, *lower_thread_priority*, *isInterruptedCall*, *isInterruptedRet*, *get_priorityLevel* $|\!\rbrace$

channelset *LockingSync* ==
$\lbrace\!|$ *lockAcquired*, *startSyncMeth*, *endSyncMeth*, *waitCall*, *waitRet*, *notify*, *isInterruptedCall*, *isInterruptedRet*,
interruptedCall, *interruptedRet*, *done_toplevel_sequencer*, *get_priorityLevel* $|\!\rbrace$

2.2 MethodCallBinder

$\textbf{channelset}\ \mathit{MethodCallBinderSync} == \{\!|\ \mathit{done\_toplevel\_sequencer},\ |\!\}$

$\textbf{process}\ \mathit{MethodCallBinder} \mathrel{\widehat=} \textbf{begin}$

$\mathit{BinderActions} \mathrel{\widehat=}$

$)\,($

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

$\textbf{end}$

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

2.3 Locking

$$\mathbf{process}\ \mathit{Threads} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{ThreadFW}(\mathit{MT2ThreadID},) \\ \interleave \\ \mathit{ThreadFW}(\mathit{MT1ThreadID},) \end{array}\right)$$

$$\mathbf{process}\ \mathit{Objects} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{ObjectFW}(\mathit{MyAppObjectID}) \\ \interleave \\ \mathit{ObjectFW}(\mathit{MissionAObjectID}) \\ \interleave \\ \mathit{ObjectFW}(\mathit{MT2ObjectID}) \\ \interleave \\ \mathit{ObjectFW}(\mathit{MissionBObjectID}) \\ \interleave \\ \mathit{ObjectFW}(\mathit{MT1ObjectID}) \end{array}\right)$$

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

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*, *MT2App*, *MissionBApp*, *MT1App*

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

$$\begin{array}{l}\textbf{process}\ Tier0 \mathrel{\widehat=} \\ \left(\begin{array}{l} MissionFW(MissionAID) \\ \quad \llbracket MissionSync \rrbracket \\ \left(ManagedThreadFW(MT2ID)\right) \end{array}\right) \\ \quad \llbracket ClusterSync \rrbracket \\ \left(\begin{array}{l} MissionFW(MissionBID) \\ \quad \llbracket MissionSync \rrbracket \\ \left(ManagedThreadFW(MT1ID)\right) \end{array}\right)\end{array}$$

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

$$\textbf{process}\ Application \mathrel{\widehat=} \left(\begin{array}{l} MyAppApp \\ \interleave \\ mainSequencerApp \\ \interleave \\ MissionAApp \\ \interleave \\ MT2App \\ \interleave \\ MissionBApp \\ \interleave \\ MT1App \end{array}\right)$$

$$\textbf{process}\ Program \mathrel{\widehat=} \left(Framework \llbracket AppSync \rrbracket ApplicationB\right) \llbracket LockingSync \rrbracket Locking$$

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

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

```
┌─ State ──────────────────────────────
│ this : ref mainSequencerClass
└──────────────────────────────────────
```

state *State*

```
┌─ Init ───────────────────────────────
│ State′
├──────────
│ this′ = new mainSequencerClass()
└──────────────────────────────────────
```

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

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

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

end

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

$$\begin{array}{|l}
\textbf{state}\ State \\
\hline
releases : \mathbb{Z} \\
\hline
\end{array}$$

state $State$

$$\begin{array}{|l}
\textbf{initial}\ Init \\
\hline
State\,' \\
\hline
releases' = 0 \\
\hline
\end{array}$$

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

$$\left(\begin{array}{l}
\textbf{if}\ (releases = 0) \longrightarrow \\
\qquad \left(\begin{array}{l}
\textbf{var}\ missionA : MissionID \bullet missionA := MissionA\,; \\
releases := releases + 1; \\
ret := missionA
\end{array}\right) \\
[\!]\ (releases = 0) \longrightarrow \\
\quad\ \ \textbf{if}\ (releases = 1) \longrightarrow \\
\qquad \left(\begin{array}{l}
\textbf{var}\ missionB : MissionID \bullet missionB := MissionB\,; \\
releases := releases + 1; \\
ret := missionB
\end{array}\right) \\
[\!]\ (releases = 1) \longrightarrow \\
\qquad \big(ret := nullMissionId\big) \\
\textbf{fi} \\
\textbf{fi}
\end{array}\right)$$

$\bullet$ **Skip**

end

5 Missions

5.1 MissionA

section $MissionAApp$ **parents** $scj\_prelude, MissionId, MissionIds,$
$SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan$
$, MissionAMethChan$

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

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

state $State$

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

$$InitializePhase \mathrel{\widehat=} \left(\begin{array}{l} initializeCall\,.\,MissionA \longrightarrow \\ register\,!\,MT2\,!\,MissionA \longrightarrow \\ initializeRet\,.\,MissionA \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

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

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

end

5.2 Schedulables of MissionA

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

process $MT2App \mathrel{\widehat=}$ **begin**

$$Run \mathrel{\widehat=} \left(\begin{array}{l} runCall \,.\, MT2 \longrightarrow \\ \left(\mathbf{Skip}\right) ; \\ runRet \,.\, MT2 \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

$\bullet\ (Methods) \mathbin{\triangle} (end\_managedThread\_app \,.\, MT2 \longrightarrow \mathbf{Skip})$

end

5.3 MissionB

section *MissionBApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*
, *MissionBMethChan*

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

```
┌─ State ─────────────────────────────
│ this : ref MissionBClass
└─────────────────────────────────────
```

state *State*

```
┌─ Init ──────────────────────────────
│ State'
├─────────────
│ this' = new MissionBClass()
└─────────────────────────────────────
```

$$\mathit{InitializePhase} \mathrel{\widehat{=}} \left(\begin{array}{l} \mathit{initializeCall}\,.\,\mathit{MissionB} \longrightarrow \\ \mathit{register}\,!\,\mathit{MT1}\,!\,\mathit{MissionB} \longrightarrow \\ \mathit{initializeRet}\,.\,\mathit{MissionB} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

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

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

end

5.4 Schedulables of MissionB

$\mathbf{section}\ MT1App\ \mathbf{parents}\ ManagedThreadChan, SchedulableId, SchedulableIds$

$\mathbf{process}\ MT1App \mathrel{\widehat=} \mathbf{begin}$

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

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

$$\bullet\ (Methods) \bigtriangleup (end\_managedThread\_app\,.\,MT1 \longrightarrow \mathbf{Skip})$$

$\mathbf{end}$