ThreeOneShotsExample

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

$$MissionAID : MissionID$$

$$distinct\langle nullMissionId, MissionAID\rangle$$

1.2 SchedulablesIds

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

$$\begin{array}{|l}
mainSequencerID : SchedulableID \\
OSEH1ID : SchedulableID \\
OSEH2ID : SchedulableID \\
OSEH3ID : SchedulableID \\
\hline
distinct \langle nullSequencerId, nullSchedulableId, mainSequencerIDID, \\
OSEH1ID, OSEH2ID, \\
OSEH3ID \rangle
\end{array}$$

1.3 ThreadIds

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

$$
\begin{array}{|l}
\mathit{OSEH2ThreadID} : \mathit{ThreadID} \\
\mathit{OSEH1ThreadID} : \mathit{ThreadID} \\
\mathit{OSEH3ThreadID} : \mathit{ThreadID} \\
\hline
\mathit{distinct} \langle \mathit{SafeletThreadId}, \mathit{nullThreadId}, \\
\mathit{OSEH2ThreadID}, \mathit{OSEH1ThreadID}, \\
\mathit{OSEH3ThreadID} \rangle
\end{array}
$$

1.4 ObjectIds

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

$$\begin{array}{|l}
\mathit{MyAppObjectID} : \mathit{ObjectID} \\
\mathit{MissionAObjectID} : \mathit{ObjectID} \\
\mathit{OSEH1ObjectID} : \mathit{ObjectID} \\
\mathit{OSEH2ObjectID} : \mathit{ObjectID} \\
\mathit{OSEH3ObjectID} : \mathit{ObjectID} \\
\hline
\mathit{distinct} \langle \mathit{MyAppObjectID}, \mathit{MissionAObjectID}, \\
\mathit{OSEH1ObjectID}, \mathit{OSEH2ObjectID}, \\
\mathit{OSEH3ObjectID} \rangle
\end{array}$$

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* ==
⋃{*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 $\mathit{MethodCallBinderSync} == \{\!|\ \mathit{done\_toplevel\_sequencer},\ |\!\}$

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

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

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

end

process $\mathit{ApplicationB} \mathrel{\widehat=} \mathit{Application} \lbrack\!\lbrack\ \mathit{MethodCallBinderSync}\ \rbrack\!\rbrack\ \mathit{MethodCallBinder}$

2.3 Locking

process *Threads* $\widehat{=}$

$$\left(\begin{array}{l} \mathit{ThreadFW}(\mathit{OSEH2ThreadID},) \\ ||| \\ \mathit{ThreadFW}(\mathit{OSEH1ThreadID},) \\ ||| \\ \mathit{ThreadFW}(\mathit{OSEH3ThreadID},) \end{array}\right)$$

process *Objects* $\widehat{=}$

$$\left(\begin{array}{l} \mathit{ObjectFW}(\mathit{MyAppObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{MissionAObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{OSEH1ObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{OSEH2ObjectID}) \\ ||| \\ \mathit{ObjectFW}(\mathit{OSEH3ObjectID}) \end{array}\right)$$

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

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*, *OSEH1App*, *OSEH2App*,
OSEH3App

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

$$\textbf{process}\ Tier0 \mathrel{\widehat=} \left(\begin{array}{l} MissionFW(MissionAID) \\ \qquad \llbracket MissionSync \rrbracket \\ \left(\begin{array}{l} \left(\begin{array}{l} OneShotEventHandlerFW(OSEH1ID) \\ \qquad \llbracket SchedulablesSync \rrbracket \\ OneShotEventHandlerFW(OSEH2ID) \\ \qquad \llbracket SchedulablesSync \rrbracket \\ OneShotEventHandlerFW(OSEH3ID) \end{array}\right) \\ \quad \llbracket SchedulablesSync \rrbracket \end{array}\right) \end{array}\right)$$

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

$$\textbf{process}\ Application \mathrel{\widehat=} \left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MissionAApp \\ ||| \\ OSEH1App(RelativeTime, AapParams, MissionAID) \\ ||| \\ OSEH2App(RelativeTime, AapParams, MissionAID) \\ ||| \\ OSEH3App(RelativeTime, AapParams, MissionAID) \end{array}\right)$$

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

3 Safelet

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

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

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

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

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

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

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

end

4 Top Level Mission Sequencer

$\mathbf{section}\ mainSequencerApp\ \mathbf{parents}\ TopLevelMissionSequencerChan,$
$\quad MissionId, MissionIds, SchedulableId, mainSequencerClass$

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

$$\begin{array}{|l}
\multicolumn{1}{l}{State} \\ \hline
this : \mathbf{ref}\ mainSequencerClass \\ \hline
\end{array}$$

$\mathbf{state}\ State$

$$\begin{array}{|l}
\multicolumn{1}{l}{Init} \\ \hline
State\,' \\ \hline
this' = \mathbf{new}\ mainSequencerClass() \\ \hline
\end{array}$$

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

$$\begin{array}{l}
Methods \mathrel{\widehat=} \\
\big(GetNextMission\big)\ ;\ Methods
\end{array}$$

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

$\mathbf{end}$

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

state *State*

$notReleased : \mathbb{B}$

state *State*

initial *Init*

$State~'$

$notReleased' = true$

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

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

$\bullet$ **Skip**

end

5 Missions

5.1 MissionA

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

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

State

$\mathit{this} : \mathbf{ref}\ \mathit{MissionAClass}$

state State

Init

$\mathit{State}\,'$

$\mathit{this}' = \mathbf{new}\ \mathit{MissionAClass}()$

$$\mathit{InitializePhase} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{initializeCall} \,.\, \mathit{MissionA} \longrightarrow \\ \mathit{register}\,!\,\mathit{OSEH}1\,!\,\mathit{MissionA} \longrightarrow \\ \mathit{register}\,!\,\mathit{OSEH}2\,!\,\mathit{MissionA} \longrightarrow \\ \mathit{register}\,!\,\mathit{OSEH}3\,!\,\mathit{MissionA} \longrightarrow \\ \mathit{initializeRet} \,.\, \mathit{MissionA} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

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

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

end

5.2 Schedulables of MissionA

section $OSEH1App$ **parents** $OneShotEventHandlerChan, SchedulableId, SchedulableIds$

process $OSEH1App \mathrel{\widehat=}$
$\quad start : HighResolutionTime,$
$controllingMission : MissionID \bullet$ **begin**

$$handleAsyncEvent \mathrel{\widehat=}$$
$$\left(\begin{array}{l} handleAsyncEventCall \,.\, OSEH1 \longrightarrow \\ \big(requestTerminationCall \,.\, controllingMision \,.\, OSEH1 \longrightarrow \quad requestTerminationRet \,.\, controllingMision \,.\, OSEH1 \,?\, reques \\ handleAsyncEventRet \,.\, OSEH1 \longrightarrow \\ \mathbf{Skip} \end{array}\right.$$

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

$$\bullet\ (Methods) \mathbin{\triangle} (end\_oneShot\_app \,.\, OSEH1 \longrightarrow \mathbf{Skip})$$

end

class $OSEH1Class \mathrel{\widehat{=}}$ **begin**

state $State$

$controllingMision : Mission$

state $State$

initial $Init$

$State'$

$\bullet$ **Skip**

end

section $OSEH2App$ **parents** $OneShotEventHandlerChan, SchedulableId, SchedulableIds$

process $OSEH2App \mathrel{\widehat=}$
$\quad start : HighResolutionTime,$
$controllingMission : MissionID \bullet$ **begin**

$$handleAsyncEvent \mathrel{\widehat=}$$
$$\left(\begin{array}{l} handleAsyncEventCall \,.\, OSEH2 \longrightarrow \\ \left(requestTerminationCall \,.\, controllingMision \,.\, OSEH2 \longrightarrow \quad requestTerminationRet \,.\, controllingMision \,.\, OSEH2 \,?\, reques \right. \\ handleAsyncEventRet \,.\, OSEH2 \longrightarrow \\ \mathbf{Skip} \end{array}\right.$$

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

$$\bullet\ (Methods) \mathrel{\triangle} (end\_oneShot\_app \,.\, OSEH2 \longrightarrow \mathbf{Skip})$$

end

class $OSEH2Class \mathrel{\widehat=}$ **begin**

state $State$

$controllingMision : Mission$

state $State$

initial $Init$

$State'$

$\bullet$ **Skip**

end

section $\mathit{OSEH3App}$ **parents** $\mathit{OneShotEventHandlerChan}, \mathit{SchedulableId}, \mathit{SchedulableIds}$

process $\mathit{OSEH3App} \mathrel{\widehat=}$
$\quad \mathit{start} : \mathit{HighResolutionTime},$
$\mathit{controllingMission} : \mathit{MissionID} \bullet$ **begin**

$$\mathit{handleAsyncEvent} \mathrel{\widehat=}$$
$$\left(\begin{array}{l}
\mathit{handleAsyncEventCall} \,.\, \mathit{OSEH3} \longrightarrow \\
\big(\mathit{requestTerminationCall} \,.\, \mathit{controllingMision} \,.\, \mathit{OSEH3} \longrightarrow \quad \mathit{requestTerminationRet} \,.\, \mathit{controllingMision} \,.\, \mathit{OSEH3} \,?\, \mathit{reques} \\
\mathit{handleAsyncEventRet} \,.\, \mathit{OSEH3} \longrightarrow \\
\mathbf{Skip}
\end{array}\right.$$

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

$$\bullet \; (\mathit{Methods}) \mathbin{\triangle} (\mathit{end\_oneShot\_app} \,.\, \mathit{OSEH3} \longrightarrow \mathbf{Skip})$$

end

class $OSEH3Class \mathrel{\widehat{=}}$ **begin**

state $State$

$controllingMision : Mission$

state $State$

initial $Init$

$State'$

$\bullet$ **Skip**

end