SimpleNestedSequencer(nestedSequencer1)

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

$$MainMissionID : MissionID$$
$$NestedMissionID : MissionID$$

$$distinct\langle nullMissionId, MainMissionID, NestedMissionID\rangle$$

1.2 SchedulablesIds

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

$$
\begin{array}{l}
MainMissionSequencerID : SchedulableID \\
NestedMissionSequencerID : SchedulableID \\
NestedOneShotEventHandlerID : SchedulableID \\
\hline
distinct\langle nullSequencerId, nullSchedulableId, MainMissionSequencerIDID, \\
NestedMissionSequencerID, NestedOneShotEventHandlerID\rangle
\end{array}
$$

1.3 ThreadIds

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

$$NestedOneShotEventHandlerThreadID : ThreadID$$
$$NestedMissionSequencerThreadID : ThreadID$$

$$distinct\langle SafeletThreadId, nullThreadId, NestedOneShotEventHandlerThreadID, NestedMissionSequencerThreadID\rangle$$

1.4 ObjectIds

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

$$
\begin{array}{l}
MySafeletObjectID : ObjectID \\
MainMissionObjectID : ObjectID \\
NestedMissionSequencerObjectID : ObjectID \\
NestedMissionObjectID : ObjectID \\
NestedOneShotEventHandlerObjectID : ObjectID \\
\hline
distinct\langle MySafeletObjectID, MainMissionObjectID, \\
NestedMissionSequencerObjectID, NestedMissionObjectID, \\
NestedOneShotEventHandlerObjectID\rangle
\end{array}
$$

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* . *MainMission*, *done_mission* . *MainMission*, *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* |}

channelset *Tier*0*Sync* ==
{| *done_toplevel_sequencer*, *done_safeletFW*, *start_mission* . *NestedMission*, *done_mission* . *NestedMission*, *initializeRet* . *NestedMission*, *requestTermination* . *NestedMission* . *MainMissionSequencer* |}

2.2 MethodCallBinder

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

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

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

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

end

process $ApplicationB \mathrel{\widehat{=}} Application \ [\![\ MethodCallBinderSync\ ]\!]\ MethodCallBinder$

2.3 Locking

process $Threads \mathrel{\widehat{=}}$
$$\left(\begin{array}{l} ThreadFW(NestedOneShotEventHandlerThreadID, 5) \\ ||| \\ ThreadFW(NestedMissionSequencerThreadID, 5) \end{array}\right)$$

process $Objects \mathrel{\widehat{=}}$
$$\left(\begin{array}{l} ObjectFW(MySafeletObjectID) \\ ||| \\ ObjectFW(MainMissionObjectID) \\ ||| \\ ObjectFW(NestedMissionSequencerObjectID) \\ ||| \\ ObjectFW(NestedMissionObjectID) \\ ||| \\ ObjectFW(NestedOneShotEventHandlerObjectID) \end{array}\right)$$

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

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*,
MySafeletApp, *MainMissionSequencerApp*, *MainMissionApp*, *NestedMissionSequencerApp*, *NestedMissionApp*, *Nested*

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

process $Tier0 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(MainMissionID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} SchedulableMissionSequencerFW(NestedMissionSequencerID) \\ \quad [\![SchedulablesSync]\!] \end{array}\right) \end{array}\right)$$

process $Tier1 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(NestedMissionID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} OneShotEventHandlerFW(NestedOneShotEventHandlerID, (NULL, nullSchedulableId)) \\ \quad [\![SchedulablesSync]\!] \end{array}\right) \end{array}\right)$$

process $Framework \mathrel{\widehat=}$
$$\left(\begin{array}{l} ControlTier \\ \quad [\![TierSync]\!] \\ \left(\begin{array}{l} Tier0 \\ \quad [\![Tier0Sync]\!] \\ Tier1 \end{array}\right) \end{array}\right)$$

process $Application \mathrel{\widehat=}$
$$\left(\begin{array}{l} MySafeletApp \\ ||| \\ MainMissionSequencerApp \\ ||| \\ MainMissionApp \\ ||| \\ NestedMissionSequencerApp \\ ||| \\ NestedMissionApp \\ ||| \\ NestedOneShotEventHandlerApp(RelativeTime) \end{array}\right)$$

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

3 Safelet

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

process $MySafeletApp \mathrel{\widehat=} \textbf{begin}$

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

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

$$immortalMemorySizeMeth \mathrel{\widehat=} \textbf{var}\ ret : \mathbb{Z} \bullet \begin{pmatrix} immortalMemorySizeCall\,.\,MySafelet \longrightarrow \\ (ret := 100000)\,; \\ immortalMemorySizeRet\,.\,MySafelet\,!\,ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

4 Top Level Mission Sequencer

section *MainMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*, *MissionId*, *MissionIds*, *SchedulableId*, *MainMissionSequencerClass*

process $MainMissionSequencerApp \mathrel{\widehat=} \textbf{begin}$

State

$this : \textbf{ref}\ MainMissionSequencerClass$

state *State*

Init

$State'$

$this' = \textbf{new}\ MainMissionSequencerClass()$

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

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

$\bullet\ (Init\ ;\ \ Methods) \triangle (end\_sequencer\_app\ .\ MainMissionSequencer \longrightarrow \textbf{Skip})$

end

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

state *State*

$$returnedMission : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$returnedMission' = false$$

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

$$\left(\begin{array}{l} \textbf{if } returnedMission = \textbf{True} \longrightarrow \\ \quad \left(ret := nullMissionId\right) \\ [] \; returnedMission = \textbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} this\,.\,returnedMission := true; \\ ret := MainMission \end{array}\right) \\ \textbf{fi} \end{array}\right)$$

• **Skip**

end

5 Missions

5.1 MainMission

section *MainMissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MainMissionMethChan*

process $MainMissionApp \mathrel{\widehat=} \textbf{begin}$

State

$this : \textbf{ref}\ MainMissionClass$

state *State*

Init

$State'$

$this' = \textbf{new}\ MainMissionClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,MainMission \longrightarrow \\ register\,!\,NestedMissionSequencer\,!\,MainMission \longrightarrow \\ initializeRet\,.\,MainMission \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\,.\,MainMission \longrightarrow \\ cleanupMissionRet\,.\,MainMission\,!\,\textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

5.2 Schedulables of MainMission

section *NestedMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*, *MissionId*, *MissionIds*, *SchedulableId*, *NestedMissionSequencerClass*

process $NestedMissionSequencerApp \mathrel{\widehat=} \textbf{begin}$

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

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

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

end

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

state *State*

$$returnedMission : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$returnedMission' = false$$

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

$$\begin{pmatrix} \textbf{if } returnedMission = \textbf{True} \longrightarrow \\ \quad (ret := nullMissionId) \\ [] \; returnedMission = \textbf{True} \longrightarrow \\ \quad \begin{pmatrix} this\,.\,returnedMission := true; \\ ret := NestedMission \end{pmatrix} \\ \textbf{fi} \end{pmatrix}$$

• **Skip**

end

5.3 NestedMission

section *NestedMissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *NestedMissionMethChan*

process $NestedMissionApp \mathrel{\widehat=} \textbf{begin}$

State

$this : \textbf{ref}\ NestedMissionClass$

state *State*

Init

$State'$

$this' = \textbf{new}\ NestedMissionClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ NestedMission \longrightarrow \\ register\ !\ NestedOneShotEventHandler\ !\ NestedMission \longrightarrow \\ initializeRet\ .\ NestedMission \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\ .\ NestedMission \longrightarrow \\ cleanupMissionRet\ .\ NestedMission\ !\ \textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ NestedMission \longrightarrow \textbf{Skip})$$

end

5.4 Schedulables of NestedMission

section *NestedOneShotEventHandlerApp* **parents** *OneShotEventHandlerChan*, *SchedulableId*, *SchedulableIds*

process *NestedOneShotEventHandlerApp* $\widehat{=}$
time : *HighResolutionTime* • **begin**

$$
\begin{pmatrix}
handleAsyncEventCall\,.\,NestedOneShotEventHandler \longrightarrow \\
(\mathbf{Skip})\,; \\
handleAsyncEventRet\,.\,NestedOneShotEventHandler \longrightarrow \\
\mathbf{Skip}
\end{pmatrix}
$$

$$
\begin{pmatrix}
handleAsyncEvent
\end{pmatrix}\,;\; Methods
$$

• $(Methods) \triangle (end\_oneShot\_app\,.\,NestedOneShotEventHandler \longrightarrow \mathbf{Skip})$

end