MultipleNestedMissions(nestedSequencer3)

Tight Rope v0.65

5th February 2016

1 ID Files

1.1 MissionIds

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

$$MainMissionID : MissionID$$
$$NestedMissionAID : MissionID$$
$$NestedMissionBID : MissionID$$

$$distinct\langle nullMissionId, MainMissionID, NestedMissionAID, NestedMissionBID\rangle$$

1.2 SchedulablesIds

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

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

1.3 ThreadIds

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

$$MT2ThreadID : ThreadID$$
$$MT1ThreadID : ThreadID$$
$$NestedMissionSequencerThreadID : ThreadID$$

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

1.4 ObjectIds

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

$$
\begin{array}{l}
MyAppObjectID : ObjectID \\
MainMissionObjectID : ObjectID \\
NestedMissionSequencerObjectID : ObjectID \\
NestedMissionAObjectID : ObjectID \\
MT1ObjectID : ObjectID \\
NestedMissionBObjectID : ObjectID \\
MT2ObjectID : ObjectID \\
\hline
distinct\langle MyAppObjectID, MainMissionObjectID, \\
NestedMissionSequencerObjectID, NestedMissionAObjectID, \\
MT1ObjectID, NestedMissionBObjectID, \\
MT2ObjectID\rangle
\end{array}
$$

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* . *NestedMissionA*, *done_mission* . *NestedMissionA*, *initializeRet* . *NestedMissionA*, *requestTermination* . *NestedMissionA* . *mainSequencer*, *start_mission* . *NestedMissionB*, *done_mission* . *NestedMissionB*, *initializeRet* . *NestedMissionB*, *requestTermination* . *NestedMissionB* . *mainSequencer* |}

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 $Threads \mathrel{\widehat{=}}$
$$\left(\begin{array}{l} ThreadFW(MT2\,ThreadID,) \\ |||\\ ThreadFW(MT1\,ThreadID,) \\ |||\\ ThreadFW(NestedMissionSequencerThreadID,) \end{array}\right)$$

process $Objects \mathrel{\widehat{=}}$
$$\left(\begin{array}{l} ObjectFW(MyAppObjectID) \\ |||\\ ObjectFW(MainMissionObjectID) \\ |||\\ ObjectFW(NestedMissionSequencerObjectID) \\ |||\\ ObjectFW(NestedMissionAObjectID) \\ |||\\ ObjectFW(MT1ObjectID) \\ |||\\ ObjectFW(NestedMissionBObjectID) \\ |||\\ ObjectFW(MT2ObjectID) \end{array}\right)$$

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*,
MyAppApp, *mainSequencerApp*, *MainMissionApp*, *NestedMissionSequencerApp*, *NestedMissionAApp*, *MT*1*App*, *Nest*

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

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

process $Tier1 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(NestedMissionAID) \\ \quad [\![MissionSync]\!] \\ (ManagedThreadFW(MT1ID)) \end{array}\right)$$
$$[\![ClusterSync]\!]$$
$$\left(\begin{array}{l} MissionFW(NestedMissionBID) \\ \quad [\![MissionSync]\!] \\ (ManagedThreadFW(MT2ID)) \end{array}\right)$$

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 $Application \mathrel{\widehat=}$
$$\left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MainMissionApp \\ ||| \\ NestedMissionSequencerApp \\ ||| \\ NestedMissionAApp \\ ||| \\ MT1App \\ ||| \\ NestedMissionBApp \\ ||| \\ MT2App \end{array}\right)$$

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

3 Safelet

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

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

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

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

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

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

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

end

4 Top Level Mission Sequencer

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

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

State

$this : \mathbf{ref}\ mainSequencerClass$

state *State*

Init

$State'$

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

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

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

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

end

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

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

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

• **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 $NestedMissionSequencerApp \mathrel{\widehat=}$
$name : String \bullet$ **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*

$releases : \mathbb{Z}$

state *State*

initial *Init*

$State'$

$releases' = 0$

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

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

• **Skip**

end

5.3 NestedMissionA

section *NestedMissionAApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *NestedMissionAMethChan*

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

State

$this : \textbf{ref}\ NestedMissionAClass$

state *State*

Init

$State'$

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,NestedMissionA \longrightarrow \\ register\,!\,MT1\,!\,NestedMissionA \longrightarrow \\ initializeRet\,.\,NestedMissionA \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

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

end

5.4 Schedulables of NestedMissionA

section $MT1App$ **parents** $ManagedThreadChan, SchedulableId, SchedulableIds$

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

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

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

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

end

5.5 NestedMissionB

section *NestedMissionBApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *NestedMissionBMethChan*

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

State

$this : \textbf{ref}\ NestedMissionBClass$

state *State*

Init

$State'$

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,NestedMissionB \longrightarrow \\ register\,!\,MT2\,!\,NestedMissionB \longrightarrow \\ initializeRet\,.\,NestedMissionB \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

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

end

5.6 Schedulables of NestedMissionB

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

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

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

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

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

end