nestedSequencer1

Tight Rope v0.88

4th March 2017

1 ID Files

1.1 MissionIds

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

$$MainMissionMID : MissionID$$
$$NestedMissionMID : MissionID$$

$$distinct\langle nullMissionId, MainMissionMID, NestedMissionMID\rangle$$

1.2 SchedulablesIds

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

$$
\begin{array}{l}
MainMissionSequencerSID : SchedulableID \\
NestedMissionSequencerSID : SchedulableID \\
NestedOneShotEventHandlerSID : SchedulableID \\
\hline
distinct\langle nullSequencerId, nullSchedulableId, MainMissionSequencerSID, \\
NestedMissionSequencerSID, NestedOneShotEventHandlerSID\rangle
\end{array}
$$

1.3 Non-Paradigm Objects

1.4 ThreadIds

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

$$SafeletTId : ThreadID$$
$$nullThreadId : ThreadID$$

$$distinct\langle SafeletTId, nullThreadId\rangle$$

1.5 ObjectIds

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

$$\vdash \; distinct\langle\rangle$$

2 Network

2.1 Network Channel Sets

section *NetworkChannels* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *TopLevelMissionSequencerFWChan*,
FrameworkChan, *SafeletChan*, *AperiodicEventHandlerChan*, *ManagedThreadChan*,
OneShotEventHandlerChan, *PeriodicEventHandlerChan*, *MissionSequencerMethChan*

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 *SafeltAppSync* $\widehat{=}$
{| *getSequencerCall*, *getSequencerRet*, *initializeApplicationCall*, *initializeApplicationRet*, *end_safelet_app* |}

channelset *MissionSequencerAppSync* ==
{| *getNextMissionCall*, *getNextMissionRet*, *end_sequencer_app* |}

channelset *MissionAppSync* ==
{| *initializeCall*, *register*, *initializeRet*, *cleanupMissionCall*, *cleanupMissionRet* |}

channelset *AppSync* ==
$\bigcup\{$*SafeltAppSync*, *MissionSequencerAppSync*, *MissionAppSync*,
MTAppSync, *OSEHSync*, *APEHSync*, *PEHSync*,
{| *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 Locking

section *NetworkLocking* **parents** *scj_prelude*, *GlobalTypes*, *FrameworkChan*, *MissionId*, *MissionIds*, *ThreadIds*, *NetworkChannels*, *ObjectFW*, *ThreadFW*, *Priority*

process *Threads* $\widehat{=}$
$(\mathbf{Skip})$

process *Objects* $\widehat{=}$
$(\mathbf{Skip})$

process *Locking* $\widehat{=}$ $(Threads \llbracket ThreadSync \rrbracket Objects) \triangle (done\_toplevel\_sequencer \longrightarrow \mathbf{Skip})$

2.3 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*

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

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

$$\textbf{process}\ Tier1 \mathrel{\widehat=} \left(\begin{array}{l} MissionFW(NestedMissionID) \\ \quad [\![MissionSync]\!] \\ (OneShotEventHandlerFW(NestedOneShotEventHandlerID, (time(5,0)), (NULL, nullSchedulableId))) \end{array}\right)$$

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

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

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

3 Safelet

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

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

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

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

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

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

end

4 Top Level Mission Sequencer

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

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

State

$this : \textbf{ref}\ MainMissionSequencerClass$

state *State*

Init

$State'$

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

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

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

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

end

section *MainMissionSequencerClass* **parents** *scj_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan* , *MethodCallBindingChannels*, *MissionId*, *MissionIds*

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

state *State*

$$returnedMission : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$returnedMission' = \textbf{False}$$

protected *getNextMission* $\widehat{=}$

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

- **Skip**

end

5 Missions

5.1 MainMission

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

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

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

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

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

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

end

5.2 Schedulables of MainMission

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

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

State

$this : \textbf{ref}\ NestedMissionSequencerClass$

state *State*

Init

$State'$

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

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

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

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

end

section *NestedMissionSequencerClass* **parents** *scj_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

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

state *State*

$$returnedMission : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$returnedMission' = false$$

protected *getNextMission* $\widehat{=}$

$$\left(\begin{array}{l} \textbf{if } returnedMission \longrightarrow \\ \quad \big(ret := nullMissionId\big) \\ [\!] \neg\, returnedMission \longrightarrow \\ \quad \left(\begin{array}{l} returnedMission := \textbf{True}; \\ ret := NestedMissionMID \end{array}\right) \\ \textbf{fi} \end{array}\right)$$

- **Skip**

end

5.3 NestedMission

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

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

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

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

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

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

end

5.4 Schedulables of NestedMission

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

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

$$
\begin{array}{l}
handleAsyncEvent \mathrel{\widehat{=}} \\
\left(\begin{array}{l}
handleAsyncEventCall\,.\,NestedOneShotEventHandlerSID \longrightarrow \\
\textbf{Skip}; \\
handleAsyncEventRet\,.\,NestedOneShotEventHandlerSID \longrightarrow \\
\textbf{Skip}
\end{array}\right)
\end{array}
$$

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

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

end