

nestedSequencer1

Tight Rope v0.88

4th March 2017

1 ID Files

1.1 MissionIds

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

<i>MainMissionMID</i> : <i>MissionID</i> <i>NestedMissionMID</i> : <i>MissionID</i>
--

<i>distinct</i> (<i>nullMissionId</i> , <i>MainMissionMID</i> , <i>NestedMissionMID</i>)
--

1.2 SchedulablesIds

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

MainMissionSequencerSID : SchedulableID

NestedMissionSequencerSID : SchedulableID

NestedOneShotEventHandlerSID : SchedulableID

*distinct⟨nullSequencerId, nullSchedulableId, MainMissionSequencerSID,
NestedMissionSequencerSID, NestedOneShotEventHandlerSID⟩*

1.3 Non-Paradigm Objects

1.4 ThreadIds

section *ThreadId* **parents** *scj_prelude, GlobalTypes*

SafeletTid : *ThreadID*
nullThreadId : *ThreadID*

distinct(*SafeletTid*, *nullThreadId*)

1.5 ObjectIds

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

$\text{distinct} \langle \rangle$

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* $\hat{=}$
{ *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 *Tier0Sync* ==
{ *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* $\hat{=}$
(**Skip**)

process *Objects* $\hat{=}$
(**Skip**)

process *Locking* $\hat{=}$ (*Threads* \llbracket *ThreadSync* \rrbracket *Objects*) \triangle (*done_toplevel_sequencer* \longrightarrow **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, NestedOneShotEventHandlerApp*

process *ControlTier* $\hat{=}$

$$\left(\begin{array}{l} \text{SafeletFW} \\ \llbracket \text{ControlTierSync} \rrbracket \\ \text{TopLevelMissionSequencerFW}(\text{MainMissionSequencer}) \end{array} \right)$$

process *Tier0* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{MainMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{SchedulableMissionSequencerFW}(\text{NestedMissionSequencerID})) \end{array} \right)$$

process *Tier1* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{NestedMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{NestedOneShotEventHandlerID}, (\text{time}(5, 0)), (\text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

process *Framework* $\hat{=}$

$$\left(\begin{array}{l} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ \left(\begin{array}{l} \text{Tier0} \\ \llbracket \text{Tier0Sync} \rrbracket \end{array} \right) \\ \text{Tier1} \end{array} \right)$$

process *Application* $\hat{=}$

$$\left(\begin{array}{l} \text{MySafeletApp} \\ \llbracket \text{AppSync} \rrbracket \\ \text{MainMissionSequencerApp} \\ \llbracket \text{AppSync} \rrbracket \\ \text{MainMissionApp} \\ \llbracket \text{AppSync} \rrbracket \\ \text{NestedMissionSequencerApp} \\ \llbracket \text{AppSync} \rrbracket \\ \text{NestedMissionApp} \\ \llbracket \text{AppSync} \rrbracket \\ \text{NestedOneShotEventHandlerApp} \end{array} \right)$$

process *Program* $\hat{=}$ $(\text{Framework} \llbracket \text{AppSync} \rrbracket \text{Application}) \llbracket \text{LockingSync} \rrbracket \text{Locking}$

3 Safelet

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

process *MySafeletApp* $\hat{=}$ **begin**

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

GetSequencer $\hat{=}$
 $\left(\begin{array}{l} \textit{getSequencerCall} \longrightarrow \\ \textit{getSequencerRet} ! \textit{MainMissionSequencerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

Methods $\hat{=}$
 $\left(\begin{array}{l} \textit{GetSequencer} \\ \square \\ \textit{InitializeApplication} \end{array} \right); \textit{Methods}$

• $(\textit{Methods}) \triangle (\textit{end_safelet_app} \longrightarrow \mathbf{Skip})$

end

4 Top Level Mission Sequencer

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

process *MainMissionSequencerApp* $\hat{=}$ **begin**

<i>State</i> <i>this</i> : ref <i>MainMissionSequencerClass</i>

state *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = new <i>MainMissionSequencerClass</i> ()

GetNextMission $\hat{=}$ **var** *ret* : *MissionID* •
 $\left(\begin{array}{l} \textit{getNextMissionCall} . \textit{MainMissionSequencerSID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{MainMissionSequencerSID} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

Methods $\hat{=}$
 $(\textit{GetNextMission}) ; \textit{Methods}$

• $(\textit{Init} ; \textit{Methods}) \triangle (\textit{end_sequencer_app} . \textit{MainMissionSequencerSID} \longrightarrow \mathbf{Skip})$

end

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

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

state <i>State</i> <i>returnedMission</i> : \mathbb{B}
--

state *State*

initial <i>Init</i> <i>State</i> '
<i>returnedMission</i> ' = False

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } returnedMission \longrightarrow \\ \quad (ret := nullMissionId) \\ \quad \square \neg returnedMission \longrightarrow \\ \quad \quad (returnedMission := \mathbf{True}; \\ \quad \quad ret := MainMissionMID) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5 Missions

5.1 MainMission

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

process *MainMissionApp* $\hat{=}$ **begin**

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{NestedMissionSequencerSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{initializeRet} . \textit{MainMissionMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \textit{cleanupMissionCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MainMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

Methods $\hat{=}$ $\left(\begin{array}{l} \textit{InitializePhase} \\ \square \\ \textit{CleanupPhase} \end{array} \right) ; \textit{Methods}$

• $(\textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MainMissionMID} \longrightarrow \mathbf{Skip})$

end

5.2 Schedulables of MainMission

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

process *NestedMissionSequencerApp* $\hat{=}$ **begin**

<i>State</i> <i>this</i> : ref <i>NestedMissionSequencerClass</i>

state *State*

<i>Init</i> <i>State'</i>
<i>this'</i> = new <i>NestedMissionSequencerClass</i> ()

GetNextMission $\hat{=}$ **var** *ret* : *MissionID* •
 $\left(\begin{array}{l} \text{getNextMissionCall} . \text{NestedMissionSequencerSID} \longrightarrow \\ \text{ret} := \text{this} . \text{getNextMission}(); \\ \text{getNextMissionRet} . \text{NestedMissionSequencerSID} ! \text{ret} \longrightarrow \\ \text{Skip} \end{array} \right)$

Methods $\hat{=}$
 $(\text{GetNextMission}) ; \text{Methods}$

• $(\text{Init} ; \text{Methods}) \triangle (\text{end_sequencer_app} . \text{NestedMissionSequencerSID} \longrightarrow \text{Skip})$

end

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

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

state <i>State</i> <i>returnedMission</i> : \mathbb{B}
--

state *State*

initial <i>Init</i> <i>State</i> '
<i>returnedMission</i> ' = <i>false</i>

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } returnedMission \longrightarrow \\ \quad (ret := nullMissionId) \\ \quad \square \neg returnedMission \longrightarrow \\ \quad \quad (returnedMission := \mathbf{True}; \\ \quad \quad ret := NestedMissionMID) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5.3 NestedMission

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

process *NestedMissionApp* $\hat{=}$ **begin**

InitializePhase $\hat{=}$

$$\left(\begin{array}{l} \text{initializeCall} . \text{NestedMissionMID} \longrightarrow \\ \text{register} ! \text{NestedOneShotEventHandlerSID} ! \text{NestedMissionMID} \longrightarrow \\ \text{initializeRet} . \text{NestedMissionMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

CleanupPhase $\hat{=}$

$$\left(\begin{array}{l} \text{cleanupMissionCall} . \text{NestedMissionMID} \longrightarrow \\ \text{cleanupMissionRet} . \text{NestedMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

Methods $\hat{=}$ $\left(\begin{array}{c} \text{InitializePhase} \\ \square \\ \text{CleanupPhase} \end{array} \right) ; \text{Methods}$

• $(\text{Methods}) \triangle (\text{end_mission_app} . \text{NestedMissionMID} \longrightarrow \mathbf{Skip})$

end

5.4 Schedulables of NestedMission

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

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

handleAsyncEvent $\hat{=}$

$$\left(\begin{array}{l} \textit{handleAsyncEventCall} . \textit{NestedOneShotEventHandlerSID} \longrightarrow \\ \mathbf{Skip}; \\ \textit{handleAsyncEventRet} . \textit{NestedOneShotEventHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

Methods $\hat{=}$
 $(\textit{handleAsyncEvent}) ; \textit{Methods}$

• $(\textit{Methods}) \triangle (\textit{end_oneShot_app} . \textit{NestedOneShotEventHandlerSID} \longrightarrow \mathbf{Skip})$

end