

nestedSequencer3

Tight Rope v0.75

12th February 2017

1 ID Files

1.1 MissionIds

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

MainMissionMID : *MissionID*

NestedMissionAMID : *MissionID*

NestedMissionBMID : *MissionID*

distinct(*nullMissionId*, *MainMissionMID*, *NestedMissionAMID*,
NestedMissionBMID)

1.2 SchedulablesIds

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

mainSequencerSID : SchedulableID

NestedMissionSequencerSID : SchedulableID

MT1SID : SchedulableID

MT2SID : SchedulableID

distinct(*nullSequencerId, nullSchedulableId, mainSequencerSID,*
NestedMissionSequencerSID, MT1SID,
MT2SID)

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*

$distinct \langle \rangle$

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 ==  
    { SafeltAppSync, MissionSequencerAppSync, MissionAppSync,  
      MTEAppSync, 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 . NestedMissionA, done_mission . NestedMissionA,  
      initializeRet . NestedMissionA, requestTermination . NestedMissionA . mainSequencer,  
      start_mission . NestedMissionB, done_mission . NestedMissionB,  
      initializeRet . NestedMissionB, requestTermination . NestedMissionB . mainSequencer }
```

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* [*ThreadSync*] *Objects*

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, MyAppApp, mainSequencerApp, MainMissionApp, NestedMissionSequencerApp, NestedMissionAApp, MT1App, Nest*

process *ControlTier* $\hat{=}$

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

process *Tier0* $\hat{=}$

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

process *Tier1* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{NestedMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{ManagedThreadFW}(\text{MT1ID})) \\ \llbracket \text{ClusterSync} \rrbracket \\ \text{MissionFW}(\text{NestedMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{ManagedThreadFW}(\text{MT2ID})) \end{array} \right)$$

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* $\hat{=}$

$$\left(\begin{array}{l} \text{MyAppApp} \\ \llbracket \text{mainSequencerApp} \rrbracket \\ \llbracket \text{MainMissionApp} \rrbracket \\ \llbracket \text{NestedMissionSequencerApp} \rrbracket \\ \llbracket \text{NestedMissionAApp} \rrbracket \\ \llbracket \text{MT1App} \rrbracket \\ \llbracket \text{NestedMissionBApp} \rrbracket \\ \llbracket \text{MT2App} \rrbracket \end{array} \right)$$

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

3 Safelet

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

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

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

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

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 *mainSequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *mainSequencerClass*, *MethodCallBindingChannels*

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

<i>State</i> <i>this</i> : ref <i>mainSequencerClass</i>
--

state *State*

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

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

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

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

end

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

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

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

state *State*

initial <i>Init</i> <i>State</i> '
<i>notReleased</i> ' = <i>true</i>

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

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

• **Skip**

end

5 Missions

5.1 MainMission

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

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

<i>State</i> <i>this</i> : ref <i>MainMissionClass</i>
--

state *State*

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

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 $\hat{=}$
 $\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MainMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MainMissionMID} \longrightarrow \mathbf{Skip})$

end

5.2 Schedulables of MainMission

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

process *NestedMissionSequencerApp* $\hat{=}$
name : *String* • **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 *State*
releases : \mathbb{Z}

state *State*

initial *Init*
State '
releases' = 0

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

$$\left(\begin{array}{l} \text{if } (releases = 0) \longrightarrow \\ \quad \left(\begin{array}{l} releases := releases + 1; \\ ret := NestedMissionAMID \end{array} \right) \\ \square \neg (releases = 0) \longrightarrow \\ \quad \text{if } (releases = 1) \longrightarrow \\ \quad \quad \left(\begin{array}{l} releases := releases + 1; \\ ret := NestedMissionBMID \end{array} \right) \\ \square \neg (releases = 1) \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \\ \text{fi} \end{array} \right)$$

• Skip

end

5.3 NestedMissionA

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

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

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

state *State*

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

InitializePhase $\hat{=}$

$$\left(\begin{array}{l} \textit{initializeCall} . \textit{NestedMissionAMID} \longrightarrow \\ \textit{register} ! \textit{MT1SID} ! \textit{NestedMissionAMID} \longrightarrow \\ \textit{initializeRet} . \textit{NestedMissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

CleanupPhase $\hat{=}$

$$\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{NestedMissionAMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{NestedMissionAMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{NestedMissionAMID} \longrightarrow \mathbf{Skip})$

end

5.4 Schedulables of NestedMissionA

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

process *MT1App* $\hat{=}$ **begin**

Run $\hat{=}$
$$\left(\begin{array}{l} \textit{runCall} . \textit{MT1SID} \longrightarrow \\ \text{)} \{ \\ \textit{runRet} . \textit{MT1SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

• $(\textit{Methods}) \triangle (\textit{end_managedThread_app} . \textit{MT1SID} \longrightarrow \mathbf{Skip})$

end

5.5 NestedMissionB

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

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

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

state *State*

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

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{NestedMissionBMID} \longrightarrow \\ \textit{register} ! \textit{MT2SID} ! \textit{NestedMissionBMID} \longrightarrow \\ \textit{initializeRet} . \textit{NestedMissionBMID} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \textbf{var } \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{NestedMissionBMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{NestedMissionBMID} ! \textbf{True} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{NestedMissionBMID} \longrightarrow \textbf{Skip})$

end

5.6 Schedulables of NestedMissionB

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

process *MT2App* $\hat{=}$ **begin**

Run $\hat{=}$

$$\left(\begin{array}{l} \text{runCall} . \text{MT2SID} \longrightarrow \\ \text{runRet} . \text{MT2SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

• $(\text{Methods}) \triangle (\text{end_managedThread_app} . \text{MT2SID} \longrightarrow \mathbf{Skip})$

end