

# ThreeTiers(nestedSequencer4)

Tight Rope v0.65

5th February 2016

## 1 ID Files

### 1.1 MissionIds

**section** *MissionIds* **parents** *scj\_prelude, MissionId*

*TopMissionID : MissionID*  
*MidMissionID : MissionID*  
*BottomMissionAID : MissionID*  
*BottomMissionBID : MissionID*

---

*distinct⟨nullMissionId, TopMissionID, MidMissionID,  
BottomMissionAID, BottomMissionBID⟩*

## 1.2 SchedulablesIds

**section** *SchedulableIds* **parents** *scj\_prelude, SchedulableId*

<i>topSequencerID : SchedulableID</i> <i>MT1ID : SchedulableID</i> <i>MidMissionSequencerID : SchedulableID</i> <i>BottomMissionSequencerAID : SchedulableID</i> <i>BottomMissionSequencerBID : SchedulableID</i> <i>MT2ID : SchedulableID</i> <i>MT3ID : SchedulableID</i>
<i>distinct⟨nullSequencerId, nullSchedulableId, topSequencerIDID,</i> <i>MT1ID, MidMissionSequencerID,</i> <i>BottomMissionSequencerAID, BottomMissionSequencerBID,</i> <i>MT2ID, MT3ID⟩</i>

### 1.3 ThreadIds

**section** *ThreadId*s **parents** *scj\_prelude*, *GlobalTypes*

*MT2ThreadId* : *ThreadId*

*MT1ThreadId* : *ThreadId*

*MT3ThreadId* : *ThreadId*

*BottomMissionSequencerBThreadId* : *ThreadId*

*BottomMissionSequencerAThreadId* : *ThreadId*

*MidMissionSequencerThreadId* : *ThreadId*

---

*distinct*(*SafeletThreadId*, *nullThreadId*,  
*MT2ThreadId*, *MT1ThreadId*,  
*MT3ThreadId*, *BottomMissionSequencerBThreadId*,  
*BottomMissionSequencerAThreadId*, *MidMissionSequencerThreadId*)

## 1.4 ObjectIds

**section** *ObjectIds* **parents** *scj\_prelude, GlobalTypes*

*MyAppObjectID* : *ObjectID*  
*TopMissionObjectID* : *ObjectID*  
*MT1ObjectID* : *ObjectID*  
*MidMissionSequencerObjectID* : *ObjectID*  
*MidMissionObjectID* : *ObjectID*  
*BottomMissionSequencerAObjectID* : *ObjectID*  
*BottomMissionSequencerBObjectID* : *ObjectID*  
*BottomMissionAObjectID* : *ObjectID*  
*MT2ObjectID* : *ObjectID*  
*BottomMissionBObjectID* : *ObjectID*  
*MT3ObjectID* : *ObjectID*

$$\text{distinct}\langle \text{MyAppObjectID}, \text{TopMissionObjectID}, \\ \text{MT1ObjectID}, \text{MidMissionSequencerObjectID}, \\ \text{MidMissionObjectID}, \text{BottomMissionSequencerAObjectID}, \\ \text{BottomMissionSequencerBObjectID}, \text{BottomMissionAObjectID}, \\ \text{MT2ObjectID}, \text{BottomMissionBObjectID}, \\ \text{MT3ObjectID} \rangle$$

## 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 . TopMission, done_mission . TopMission,  
      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 ==  
    { 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 Tier0Sync ==  
    { done_toplevel_sequencer, done_safeletFW,  
      start_mission . MidMission, done_mission . MidMission,  
      initializeRet . MidMission, requestTermination . MidMission . topSequencer }  
  
channelset Tier1Sync ==  
    { done_toplevel_sequencer, done_safeletFW,  
      start_mission . BottomMissionA, done_mission . BottomMissionA,  
      initializeRet . BottomMissionA, requestTermination . BottomMissionA . }  
  
channelset Tier2Sync ==  
    { done_toplevel_sequencer, done_safeletFW,  
      start_mission . BottomMissionB, done_mission . BottomMissionB,  
      initializeRet . BottomMissionB, requestTermination . BottomMissionB . }
```

## 2.2 MethodCallBinder

**channelset** *MethodCallBinderSync* == { *done\_toplevel\_sequencer*, }

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

*BinderActions*  $\hat{=}$   
) (

- *BinderActions*  $\triangle$  (*done\_toplevel\_sequencer*  $\longrightarrow$  **Skip**)

**end**

**process** *ApplicationB*  $\hat{=}$  *Application* [ *MethodCallBinderSync* ] *MethodCallBinder*

## 2.3 Locking

**process** *Threads*  $\hat{=}$

$$\left( \begin{array}{l} \text{ThreadFW}(\text{MT2ThreadID},) \\ ||| \\ \text{ThreadFW}(\text{MT1ThreadID},) \\ ||| \\ \text{ThreadFW}(\text{MT3ThreadID},) \\ ||| \\ \text{ThreadFW}(\text{BottomMissionSequencerBThreadID},) \\ ||| \\ \text{ThreadFW}(\text{BottomMissionSequencerAThreadID},) \\ ||| \\ \text{ThreadFW}(\text{MidMissionSequencerThreadID},) \end{array} \right)$$

**process** *Objects*  $\hat{=}$

$$\left( \begin{array}{l} \text{ObjectFW}(\text{MyAppObjectID}) \\ ||| \\ \text{ObjectFW}(\text{TopMissionObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MT1ObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MidMissionSequencerObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MidMissionObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionSequencerAObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionSequencerBObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionAObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MT2ObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionBObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MT3ObjectID}) \end{array} \right)$$

**process** *Locking*  $\hat{=}$  *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, topSequencerApp, TopMissionApp, MT1App, MidMissionSequencerApp, MidMissionApp, BottomMissionSequencerAApp, BottomMissionSequencerBApp, BottomMissionAApp, MT2App, BottomMissionBApp, MT3App*

**process** *ControlTier*  $\hat{=}$   

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

**process** *Tier0*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{TopMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{ManagedThreadFW}(\text{MT1ID}) \\ \text{SchedulableMissionSequencerFW}(\text{MidMissionSequencerID}) \end{array} \right) \\ \llbracket \text{SchedulablesSync} \rrbracket \end{array} \right)$$

**process** *Tier1*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{MidMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \left( \begin{array}{l} \text{SchedulableMissionSequencerFW}(\text{BottomMissionSequencerAID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \end{array} \right) \right) \\ \text{SchedulableMissionSequencerFW}(\text{BottomMissionSequencerBID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \end{array} \right)$$

**process** *Tier2*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \text{ManagedThreadFW}(\text{MT2ID}) \end{array} \right)$$

**process** *Tier3*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \text{ManagedThreadFW}(\text{MT3ID}) \end{array} \right)$$

**process** *Framework*  $\hat{=}$   

$$\left( \begin{array}{l} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ \left( \begin{array}{l} \text{Tier0} \\ \llbracket \text{Tier0Sync} \rrbracket \end{array} \right) \\ \text{Tier1} \\ \llbracket \text{Tier1Sync} \rrbracket \\ \text{Tier2} \\ \llbracket \text{Tier2Sync} \rrbracket \\ \text{Tier3} \end{array} \right)$$



$$\text{process } Application \hat{=} \left( \begin{array}{l} MyAppApp \\ ||| \\ topSequencerApp \\ ||| \\ TopMissionApp \\ ||| \\ MT1App \\ ||| \\ MidMissionSequencerApp \\ ||| \\ MidMissionApp \\ ||| \\ BottomMissionSequencerAApp \\ ||| \\ BottomMissionSequencerBApp \\ ||| \\ BottomMissionAApp \\ ||| \\ MT2App \\ ||| \\ BottomMissionBApp \\ ||| \\ MT3App \end{array} \right)$$

$$\text{process } Program \hat{=} (Framework \parallel AppSync \parallel ApplicationB) \parallel LockingSync \parallel Locking$$

### 3 Safelet

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

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

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

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

*immortalMemorySizeMeth*  $\hat{=}$  **var** *ret* :  $\mathbb{Z}$  •  
 $\left( \begin{array}{l} \textit{immortalMemorySizeCall} . \textit{MyApp} \longrightarrow \\ (\textit{ret} := \textit{Const.IMMORTAL\_MEM\_DEFAULT}) ; \\ \textit{immortalMemorySizeRet} . \textit{MyApp} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

• (*Methods*)  $\triangle$  (*end\_safelet\_app*  $\longrightarrow$  **Skip**)

**end**

## 4 Top Level Mission Sequencer

**section** *topSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *topSequencerClass*

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

<i>State</i> <i>this</i> : <b>ref</b> <i>topSequencerClass</i>
---

**state** *State*

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

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

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

•  $(\textit{Init} ; \textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{topSequencer} \longrightarrow \textbf{Skip})$

**end**

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

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

**state** *State*

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

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

$\left( \begin{array}{l} \text{if } \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad \left( \begin{array}{l} \mathbf{var } \textit{mission} : \textit{MissionID} \bullet \textit{mission} := \textit{TopMission}; \\ \textit{this} . \textit{notReleased} := \textit{false}; \\ \textit{ret} := \textit{mission} \end{array} \right) \\ \parallel \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad (\textit{ret} := \textit{nullMissionId}) \\ \text{fi} \end{array} \right)$

• **Skip**

**end**

## 5 Missions

### 5.1 TopMission

**section** *TopMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*,  
*TopMissionMethChan*

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

<i>State</i> <i>this</i> : <b>ref</b> <i>TopMissionClass</i>
---

**state** *State*

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

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{TopMission} \longrightarrow \\ \textit{register} ! \textit{MT1} ! \textit{TopMission} \longrightarrow \\ \textit{register} ! \textit{MidMissionSequencer} ! \textit{TopMission} \longrightarrow \\ \textit{initializeRet} . \textit{TopMission} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

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

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_mission\_app* . *TopMission*  $\longrightarrow$  **Skip**)

**end**

## 5.2 Schedulables of TopMission

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

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

*Run*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{runCall} . \textit{MT1} \longrightarrow \\ (\mathbf{Skip}) ; \\ \textit{runRet} . \textit{MT1} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

**end**

**section** *MidMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *MidMissionSequencerClass*

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

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

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

•  $(\textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{MidMissionSequencer} \longrightarrow \mathbf{Skip})$

**end**

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

---

**state** *State*  
*notReleased* :  $\mathbb{B}$

---

**state** *State*

---

**initial** *Init*  
*State* '  


---

*notReleased*' = *true*

---

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

$\left( \begin{array}{l} \text{if } \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad \left( \begin{array}{l} \mathbf{var } \textit{mission} : \textit{MissionID} \bullet \textit{mission} := \textit{MidMission}; \\ \textit{this} . \textit{notReleased} := \textit{false}; \\ \textit{ret} := \textit{mission} \end{array} \right) \\ \parallel \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad (\textit{ret} := \textit{nullMissionId}) \\ \mathbf{fi} \end{array} \right)$

• **Skip**

**end**



### 5.3 MidMission

**section** *MidMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*,  
*MidMissionMethChan*

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

<i>State</i> <i>this</i> : <b>ref</b> <i>MidMissionClass</i>
---

**state** *State*

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

*InitializePhase*  $\hat{=}$   

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

*CleanupPhase*  $\hat{=}$   

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

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_mission\_app* . *MidMission*  $\longrightarrow$  **Skip**)

**end**

## 5.4 Schedulables of MidMission

**section** *BottomMissionSequencerAApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *BottomMissionSequencerAClass*

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

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

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

•  $(\textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{BottomMissionSequencerA} \longrightarrow \mathbf{Skip})$

**end**

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

---

**state** *State*  
*notReleased* :  $\mathbb{B}$

---

**state** *State*

---

**initial** *Init*  
*State* '  


---

*notReleased*' = *true*

---

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •  
 $\left( \begin{array}{l} \text{if } \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad \left( \begin{array}{l} \mathbf{var } \textit{mission} : \textit{MissionID} \bullet \textit{mission} := \textit{BottomMissionA}; \\ \textit{this} . \textit{notReleased} := \textit{false}; \\ \textit{ret} := \textit{mission} \end{array} \right) \\ \parallel \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad (\textit{ret} := \textit{nullMissionId}) \\ \text{fi} \end{array} \right)$

• **Skip**

**end**

**section** *BottomMissionSequencerBApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *BottomMissionSequencerBClass*

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

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

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

•  $(\textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{BottomMissionSequencerB} \longrightarrow \mathbf{Skip})$

**end**

```
class BottomMissionSequencerBClass  $\hat{=}$  begin
```

---

```
  state State  
    notReleased :  $\mathbb{B}$ 
```

---

```
state State
```

---

```
  initial Init  
    State '  
    notReleased' = true
```

---

```
protected getNextMission  $\hat{=}$  var ret : MissionID •  
   $\left( \begin{array}{l} \text{if } \textit{notReleased} = \textbf{True} \longrightarrow \\ \quad \left( \textbf{var } \textit{mission} : \textit{MissionID} \bullet \textit{mission} := \textit{BottomMissionB}; \right. \\ \quad \quad \left. \textit{this} . \textit{notReleased} := \textit{false}; \right. \\ \quad \quad \left. \textit{ret} := \textit{mission} \right) \\ \quad \text{[] } \textit{notReleased} = \textbf{True} \longrightarrow \\ \quad \quad \left( \textit{ret} := \textit{nullMissionId} \right) \\ \text{fi} \end{array} \right)$ 
```

```
  • Skip
```

```
end
```

## 5.5 BottomMissionA

**section** *BottomMissionAApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, BottomMissionAMethChan*

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

<i>State</i> <i>this</i> : <b>ref</b> <i>BottomMissionAClass</i>
---

**state** *State*

<i>Init</i> <i>State'</i> <i>this'</i> = <b>new</b> <i>BottomMissionAClass</i> ()
---

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{BottomMissionA} \longrightarrow \\ \textit{register} ! \textit{MT2} ! \textit{BottomMissionA} \longrightarrow \\ \textit{initializeRet} . \textit{BottomMissionA} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_mission\_app* . *BottomMissionA*  $\longrightarrow$  **Skip**)

**end**

## 5.6 Schedulables of BottomMissionA

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

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

*Run*  $\hat{=}$   
 $\left( \begin{array}{l} \text{runCall} . MT2 \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . MT2 \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

**end**

## 5.7 BottomMissionB

**section** *BottomMissionBApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, BottomMissionBMethChan*

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

<i>State</i> <i>this</i> : <b>ref</b> <i>BottomMissionBClass</i>
---

**state** *State*

<i>Init</i> <i>State'</i> <i>this'</i> = <b>new</b> <i>BottomMissionBClass</i> ()
---

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{BottomMissionB} \longrightarrow \\ \textit{register} ! \textit{MT3} ! \textit{BottomMissionB} \longrightarrow \\ \textit{initializeRet} . \textit{BottomMissionB} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_mission\_app* . *BottomMissionB*  $\longrightarrow$  **Skip**)

**end**



## 5.8 Schedulables of BottomMissionB

**section** *MT3App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds*

**process** *MT3App*  $\hat{=}$  **begin**

*Run*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{runCall} . \textit{MT3} \longrightarrow \\ (\mathbf{Skip}) ; \\ \textit{runRet} . \textit{MT3} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Methods}) \triangle (\textit{end\_managedThread\_app} . \textit{MT3} \longrightarrow \mathbf{Skip})$

**end**