

# ThreeTiersExtension(nestedSequencer5)

Tight Rope v0.65

5th February 2016

## 1 ID Files

### 1.1 MissionIds

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

*TopMissionID : MissionID*  
*MidMissionAID : MissionID*  
*BottomMissionAID : MissionID*  
*MidMissionBID : MissionID*  
*BottomMissionBID : MissionID*

---

*distinct⟨nullMissionId, TopMissionID, MidMissionAID,  
BottomMissionAID, MidMissionBID,  
BottomMissionBID⟩*

## 1.2 SchedulablesIds

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

|   |
|---|
| <i>topSequencerID : SchedulableID</i><br><i>MT1ID : SchedulableID</i><br><i>MidMissionSequencerID : SchedulableID</i><br><i>BottomMissionSequencerAID : SchedulableID</i><br><i>OSEHID : SchedulableID</i><br><i>MT2ID : SchedulableID</i><br><i>BottomMissionSequencerBID : SchedulableID</i><br><i>APEHID : SchedulableID</i><br><i>PEHID : SchedulableID</i> |
| <i>distinct (nullSequencerId, nullSchedulableId, topSequencerIDID,</i><br><i>MT1ID, MidMissionSequencerID,</i><br><i>BottomMissionSequencerAID, OSEHID,</i><br><i>MT2ID, BottomMissionSequencerBID,</i><br><i>APEHID, PEHID)</i>  |

### 1.3 ThreadIds

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

*MT2ThreadId* : *ThreadId*  
*MT1ThreadId* : *ThreadId*  
*BottomMissionSequencerBThreadId* : *ThreadId*  
*PEHThreadId* : *ThreadId*  
*APEHThreadId* : *ThreadId*  
*BottomMissionSequencerAThreadId* : *ThreadId*  
*MidMissionSequencerThreadId* : *ThreadId*  
*OSEHThreadId* : *ThreadId*

---

*distinct*(*SafeletThreadId*, *nullThreadId*,  
*MT2ThreadId*, *MT1ThreadId*,  
*BottomMissionSequencerBThreadId*, *PEHThreadId*,  
*APEHThreadId*, *BottomMissionSequencerAThreadId*,  
*MidMissionSequencerThreadId*, *OSEHThreadId*)

## 1.4 ObjectIds

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

*MyAppObjectID* : *ObjectID*  
*TopMissionObjectID* : *ObjectID*  
*MT1ObjectID* : *ObjectID*  
*MidMissionSequencerObjectID* : *ObjectID*  
*MidMissionAObjectID* : *ObjectID*  
*BottomMissionSequencerAObjectID* : *ObjectID*  
*BottomMissionAObjectID* : *ObjectID*  
*OSEHObjectID* : *ObjectID*  
*MT2ObjectID* : *ObjectID*  
*MidMissionBObjectID* : *ObjectID*  
*BottomMissionSequencerBObjectID* : *ObjectID*  
*BottomMissionBObjectID* : *ObjectID*  
*APEHObjectID* : *ObjectID*  
*PEHObjectID* : *ObjectID*

---

*distinct*(*MyAppObjectID*, *TopMissionObjectID*,  
*MT1ObjectID*, *MidMissionSequencerObjectID*,  
*MidMissionAObjectID*, *BottomMissionSequencerAObjectID*,  
*BottomMissionAObjectID*, *OSEHObjectID*,  
*MT2ObjectID*, *MidMissionBObjectID*,  
*BottomMissionSequencerBObjectID*, *BottomMissionBObjectID*,  
*APEHObjectID*, *PEHObjectID*)

## 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 . MidMissionA, done\_mission . MidMissionA, initializeRet . MidMissionA, requestTermination . MidMissionA . topSequencer* }

**channelset** *Tier1Sync* ==  
    { *done\_toplevel\_sequencer, done\_safeletFW, start\_mission . BottomMissionA, done\_mission . BottomMissionA, initializeRet . BottomMissionA, requestTermination . BottomMissionA . , start\_mission . MidMissionB, done\_mission . MidMissionB, initializeRet . MidMissionB, requestTermination . MidMissionB . }*

```

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*  $\hat{=}$

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

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

$$\left( \begin{array}{l} \text{ObjectFW}(\text{MyAppObjectID}) \\ ||| \\ \text{ObjectFW}(\text{TopMissionObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MT1ObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MidMissionSequencerObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MidMissionAObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionSequencerAObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionAObjectID}) \\ ||| \\ \text{ObjectFW}(\text{OSEHObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MT2ObjectID}) \\ ||| \\ \text{ObjectFW}(\text{MidMissionBObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionSequencerBObjectID}) \\ ||| \\ \text{ObjectFW}(\text{BottomMissionBObjectID}) \\ ||| \\ \text{ObjectFW}(\text{APEHObjectID}) \\ ||| \\ \text{ObjectFW}(\text{PEHObjectID}) \end{array} \right)$$

**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*  
*, MidMissionAApp, BottomMissionSequencerAApp, BottomMissionAApp, MT2App, OSEHApp*  
*, MidMissionBApp, BottomMissionSequencerBApp, BottomMissionBApp, APEHApp, PEHApp*

**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*  $\hat{=}$   

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

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{ManagedThreadFW}(\text{MT2ID}) \\ \text{OneShotEventHandlerFW}(\text{OSEHID}) \end{array} \right) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \llbracket \text{ClusterSync} \rrbracket \\ \left( \begin{array}{l} \text{MissionFW}(\text{MidMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{SchedulableMissionSequencerFW}(\text{BottomMissionSequencerBID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \end{array} \right) \end{array} \right) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{AperiodicEventHandlerFW}(\text{APEHID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{PEHID}) \end{array} \right) \end{array} \right)$$

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

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

$$\text{process } Application \hat{=} \left( \begin{array}{l} MyAppApp \\ ||| \\ topSequencerApp \\ ||| \\ TopMissionApp \\ ||| \\ MT1App \\ ||| \\ MidMissionSequencerApp \\ ||| \\ MidMissionAApp \\ ||| \\ BottomMissionSequencerAApp \\ ||| \\ BottomMissionAApp \\ ||| \\ MT2App \\ ||| \\ OSEHApp(RelativeTime, AaParams, BottomMissionAID) \\ ||| \\ MidMissionBApp \\ ||| \\ BottomMissionSequencerBApp \\ ||| \\ BottomMissionBApp \\ ||| \\ APEHApp(AapParams, BottomMissionBID) \\ ||| \\ PEHApp(ApParams, afehID) \end{array} \right)$$

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

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

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

*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><br><i>this</i> : <b>ref</b> <i>topSequencerClass</i> |
|---|

**state** *State*

|  |
|--|
| <i>Init</i><br><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><br><i>notReleased</i> : $\mathbb{B}$ |
|--|

**state** *State*

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

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

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

• **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><br><i>this</i> : <b>ref</b> <i>TopMissionClass</i> |
|---|

**state** *State*

|   |
|---|
| <i>Init</i><br><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*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{runCall} . MT1 \longrightarrow \\ (\mathbf{Skip}) ; \\ \textit{runRet} . MT1 \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

**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*  
*releases* :  $\mathbb{Z}$

---

**state** *State*

---

**initial** *Init*  
*State* '  
*releases*' = 0

---

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

$$\left( \begin{array}{l} \text{if } (\text{releases} = 0) \longrightarrow \\ \quad \left( \begin{array}{l} \text{var } \text{missionA} : \text{MissionID} \bullet \text{missionA} := \text{MidMissionA}; \\ \text{releases} := \text{releases} + 1; \\ \text{ret} := \text{missionA} \end{array} \right) \\ \parallel (\text{releases} = 0) \longrightarrow \\ \quad \text{if } (\text{releases} = 1) \longrightarrow \\ \quad \quad \left( \begin{array}{l} \text{var } \text{missionB} : \text{MissionID} \bullet \text{missionB} := \text{MidMissionB}; \\ \text{releases} := \text{releases} + 1; \\ \text{ret} := \text{missionB} \end{array} \right) \\ \parallel (\text{releases} = 1) \longrightarrow \\ \quad (\text{ret} := \text{nullMissionId}) \\ \text{fi} \\ \text{fi} \end{array} \right)$$

• Skip

**end**

### 5.3 MidMissionA

**section** *MidMissionAApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MidMissionAMethChan*

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

|  |
|--|
| <i>State</i><br><i>this</i> : <b>ref</b> <i>MidMissionAClass</i> |
|--|

**state** *State*

|  |
|--|
| <i>Init</i><br><i>State'</i>                         |
| <i>this'</i> = <b>new</b> <i>MidMissionAClass</i> () |

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

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

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

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

**end**

## 5.4 Schedulables of MidMissionA

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

```
  • Skip
```

```
end
```

## 5.5 BottomMissionA

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

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

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

**state** *State*

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

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

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

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

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

**end**

**section** *OSEHApp* **parents** *OneShotEventHandlerChan*, *SchedulableId*, *SchedulableIds*

**process** *OSEHApp*  $\hat{=}$   
     *start* : *HighResolutionTime*,  
     *controllingMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{handleAsyncEventCall} . \textit{OSEH} \longrightarrow \\ \textit{requestTerminationCall} . \textit{controllingMission} . \textit{OSEH} \longrightarrow \textit{requestTerminationRet} . \textit{controllingMission} . \textit{OSEH} ? \textit{requestT} \\ \textit{handleAsyncEventRet} . \textit{OSEH} \longrightarrow \\ \mathbf{Skip} \end{array} \right.$

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

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

**end**

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

**state** *State*

*controllingMission* : *Mission*

**state** *State*

**initial** *Init*

*State'*

• **Skip**

**end**



## 5.7 MidMissionB

**section** *MidMissionBApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MidMissionBMethChan*

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

|  |
|--|
| <i>State</i><br><i>this</i> : <b>ref</b> <i>MidMissionBClass</i> |
|--|

**state** *State*

|  |
|--|
| <i>Init</i><br><i>State'</i>                         |
| <i>this'</i> = <b>new</b> <i>MidMissionBClass</i> () |

*InitializePhase*  $\hat{=}$   

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

*CleanupPhase*  $\hat{=}$   

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

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

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

**end**

## 5.8 Schedulables of MidMissionB

**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
```

## 5.9 BottomMissionB

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

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

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

**state** *State*

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

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

*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**

## 5.10 Schedulables of BottomMissionB

**section** *APEHApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*

**process** *APEHApp*  $\hat{=}$   
     *controllingMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{handleAsyncEventCall} . \textit{APEH} \longrightarrow \\ (\textit{requestTerminationCall} . \textit{controllingMission} . \textit{APEH} \longrightarrow \textit{requestTerminationRet} . \textit{controllingMission} . \textit{APEH} ? \textit{requestTerminationRet}) \\ \textit{handleAsyncEventRet} . \textit{APEH} \longrightarrow \\ \mathbf{Skip} \end{array} \right.$

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

• (*Methods*)  $\triangle$  (*end\_aperiodic\_app* . *APEH*  $\longrightarrow$  **Skip**)

**end**

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

**state** *State*

*controllingMission* : *Mission*

**state** *State*

**initial** *Init*

*State'*

• **Skip**

**end**

**section** *PEHApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*

**process** *PEHApp*  $\hat{=}$   
     *apeh* : *SchedulableID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{handleAsyncEventCall} . \textit{PEH} \longrightarrow \\ (\textit{releaseCall} . \textit{apeh} . \textit{PEH} \longrightarrow \textit{releaseRet} . \textit{apeh} . \textit{PEH} ? \textit{release} \longrightarrow \textbf{Skip}) ; \\ \textit{handleAsyncEventRet} . \textit{PEH} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

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

• (*Methods*)  $\triangle$  (*end\_periodic\_app* . *PEH*  $\longrightarrow$  **Skip**)

**end**

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

**state** *State*

*apeh* : *AperiodicEventHandler*

**state** *State*

**initial** *Init*

*State'*

• **Skip**

**end**