

nestedSequencer2 Report

4th October 2016

1 ID Files

1.1 MissionIds

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

TopMission1MID : *MissionID*

MyMission1MID : *MissionID*

MyMission2MID : *MissionID*

MyMission3MID : *MissionID*

distinct(*nullMissionId*, *TopMission1MID*, *MyMission1MID*,
MyMission2MID, *MyMission3MID*)

1.2 SchedulablesIds

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

MySequencerSID : SchedulableID

FirstMissionSequencerSID : SchedulableID

SecondMissionSequencerSID : SchedulableID

ThirdMissionSequencerSID : SchedulableID

MyPEH1SID : SchedulableID

MyPEH2SID : SchedulableID

MyPEH3SID : SchedulableID

*distinct⟨nullSequencerId, nullSchedulableId, MySequencerSID,
FirstMissionSequencerSID, SecondMissionSequencerSID,
ThirdMissionSequencerSID, MyPEH1SID,
MyPEH2SID, MyPEH3SID⟩*

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 . TopMission1, done_mission . TopMission1, 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* ==
 $\bigcup \{ \textit{SafeltAppSync, MissionSequencerAppSync, MissionAppSync, MTAppSync, OSEHSync, APEHSync, PEHSync,} \}$
 { *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 . MyMission1, done_mission . MyMission1,
    initializeRet . MyMission1, requestTermination . MyMission1 . MySequencer }

```

```

channelset Tier1Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . MyMission2, done_mission . MyMission2,
    initializeRet . MyMission2, requestTermination . MyMission2 . }

```

```

channelset Tier2Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . MyMission3, done_mission . MyMission3,
    initializeRet . MyMission3, requestTermination . MyMission3 . }

```

2.2 Locking

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

process *Threads* $\hat{=}$
 $\left(\begin{array}{l} \textit{ThreadFW}(\textit{FirstMissionSequencerTID}, 5) \\ ||| \\ \textit{ThreadFW}(\textit{MyPEH3TID}, 10) \\ ||| \\ \textit{ThreadFW}(\textit{ThirdMissionSequencerTID}, 10) \\ ||| \\ \textit{ThreadFW}(\textit{MyPEH1TID}, 5) \\ ||| \\ \textit{ThreadFW}(\textit{MyPEH2TID}, 20) \\ ||| \\ \textit{ThreadFW}(\textit{SecondMissionSequencerTID}, 15) \end{array} \right)$

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, MySequencerApp, TopMission1App, FirstMissionSequencerApp, SecondMissionSequencerApp, ThirdMissionSequencerApp, MyMission1App, MyPEH1App, MyMission2App, MyPEH2App, MyMission3App, MyPEH3App*

process *ControlTier* $\hat{=}$

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

process *Tier0* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{TopMission1ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left(\begin{array}{l} \text{OneShotEventHandlerFW}(\text{FirstMissionSequencerID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{SchedulableMissionSequencerFW}(\text{SecondMissionSequencerID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{SchedulableMissionSequencerFW}(\text{ThirdMissionSequencerID}) \end{array} \right) \end{array} \right)$$

process *Tier1* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{MyMission1ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{MyPEH1ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

process *Tier2* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{MyMission2ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{MyPEH2ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

process *Tier3* $\hat{=}$

$$\left(\begin{array}{l} \text{MissionFW}(\text{MyMission3ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{MyPEH3ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

process *Framework* $\hat{=}$

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

$$\text{process } Application \hat{=} \left(\begin{array}{l} MyAppApp \\ ||| \\ MySequencerApp \\ ||| \\ TopMission1App \\ ||| \\ FirstMission.SequencerApp \\ ||| \\ SecondMission.SequencerApp \\ ||| \\ ThirdMission.SequencerApp \\ ||| \\ MyMission1App \\ ||| \\ MyPEH1App(MyMission1ID) \\ ||| \\ MyMission2App \\ ||| \\ MyPEH2App(MyMission2ID) \\ ||| \\ MyMission3App \\ ||| \\ MyPEH3App(MyMission3ID) \end{array} \right)$$

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

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

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 *MySequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MySequencerClass*, *MethodCallBindingChannels*

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

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

state *State*

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

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

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

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

end

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

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

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

state *State*

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

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

$$\left(\begin{array}{l} \text{if } done = \mathbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} done := \mathbf{False}; \\ ret := TopMission1MID \end{array} \right) \\ \quad \square \neg done = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5 Missions

5.1 TopMission1

section *TopMission1App* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *TopMission1MethChan*,
MethodCallBindingChannels

process *TopMission1App* $\hat{=}$ **begin**

InitializePhase $\hat{=}$

$$\left(\begin{array}{l} \textit{initializeCall} . \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{FirstMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{SecondMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{ThirdMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{initializeRet} . \textit{TopMission1MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

CleanupPhase $\hat{=}$

$$\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{TopMission1MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{TopMission1MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{TopMission1MID} \longrightarrow \mathbf{Skip})$

end

5.2 Schedulables of TopMission1

section *FirstMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *FirstMissionSequencerClass*, *MethodCallBindingChannels*

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

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

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

• $(\text{Methods}) \triangle (\text{end_sequencer_app} . \text{FirstMissionSequencerSID} \longrightarrow \mathbf{Skip})$

end

```
section FirstMissionSequencerClass parents scj_prelude, SchedulableId, SchedulableIds, SafeletChan  
, MethodCallBindingChannels, MissionId, MissionIds
```

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

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

```
state State
```

<pre>initial <i>Init</i> <i>State</i> ' <i>done</i> = False</pre>

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

$$\left(\begin{array}{l} \text{if } done = \mathbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} done := \mathbf{False}; \\ ret := MyMission1MID \end{array} \right) \\ \quad \square \neg done = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

```
• Skip
```

```
end
```

section *SecondMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *SecondMissionSequencerClass*, *MethodCallBindingChannels*

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

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

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

• $(\textit{Methods}) \triangle (\textit{end_sequencer_app} . \textit{SecondMissionSequencerSID} \longrightarrow \mathbf{Skip})$

end

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

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

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

state *State*

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

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

$$\left(\begin{array}{l} \text{if } done = \mathbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} done := \mathbf{False}; \\ ret := MyMission2MID \end{array} \right) \\ \quad \square \neg done = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

section *ThirdMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,
MissionId, *MissionIds*, *SchedulableId*, *SchedulableIds*, *ThirdMissionSequencerClass*, *MethodCallBindingChannels*

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

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

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

• $(\textit{Methods}) \triangle (\textit{end_sequencer_app} . \textit{ThirdMissionSequencerSID} \longrightarrow \mathbf{Skip})$

end

```
section ThirdMissionSequencerClass parents scj_prelude, SchedulableId, SchedulableIds, SafeletChan  
, MethodCallBindingChannels, MissionId, MissionIds
```

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

state <i>State</i> _____ <i>done</i> : \mathbb{B}

```
state State
```

initial <i>Init</i> _____ <i>State</i> '
<i>done</i> = False

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

$$\left(\begin{array}{l} \text{if } done = \mathbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} done := \mathbf{False}; \\ ret := MyMission3MID \end{array} \right) \\ \quad \square \neg done = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

```
• Skip
```

```
end
```

5.3 MyMission1

section *MyMission1App* **parents** *scj_prelude, MissionId, MissionIds,*
SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission1MethChan
, MethodCallBindingChannels

process *MyMission1App* $\hat{=}$ **begin**

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MyMission1MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH1SID} ! \textit{MyMission1MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission1MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MyMission1MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission1MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MyMission1MID} \longrightarrow \mathbf{Skip})$

end

5.4 Schedulables of MyMission1

section *MyPEH1App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

process *MyPEH1App* $\hat{=}$
 m : *MissionID* • **begin**

state <i>State</i> <i>count</i> : \mathbb{Z}
--

state *State*

initial <i>Init</i> <i>State</i> '
<i>count</i> = 0

handleAsyncEvent $\hat{=}$

$$\left(\begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH1SID} \longrightarrow \\ \left(\begin{array}{l} \text{count} := \text{count} + 1; \\ \text{if } (\text{count} = 10) \longrightarrow \\ \quad \text{requestTerminationCall} . m . \text{MyPEH1SID} \longrightarrow \\ \quad \text{requestTerminationRet} . m . \text{MyPEH1SID} ? \text{requestTermination} \longrightarrow \\ \quad \text{Skip} \\ \square \neg (\text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{MyPEH1SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Methods*) \triangle (*end_periodic_app* . *MyPEH1SID* \longrightarrow **Skip**)

end

5.5 MyMission2

section *MyMission2App* **parents** *scj_prelude, MissionId, MissionIds,*
SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission2MethChan
, MethodCallBindingChannels

process *MyMission2App* $\hat{=}$ **begin**

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MyMission2MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH2SID} ! \textit{MyMission2MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission2MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MyMission2MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission2MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MyMission2MID} \longrightarrow \mathbf{Skip})$

end

5.6 Schedulables of MyMission2

section *MyPEH2App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

process *MyPEH2App* $\hat{=}$
 m : *MissionID* • **begin**

state *State* _____
 count : \mathbb{Z}

state *State*

initial *Init* _____
 State '

 count = 0

handleAsyncEvent $\hat{=}$

$$\left(\begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH2SID} \longrightarrow \\ \left(\begin{array}{l} \text{count} := \text{count} + 1; \\ \text{if } (\text{count} = 10) \longrightarrow \\ \quad \text{requestTerminationCall} . m . \text{MyPEH2SID} \longrightarrow \\ \quad \text{requestTerminationRet} . m . \text{MyPEH2SID} ? \text{requestTermination} \longrightarrow \\ \quad \text{Skip} \\ \square \neg (\text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{MyPEH2SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Methods*) \triangle (*end_periodic_app* . *MyPEH2SID* \longrightarrow **Skip**)

end

5.7 MyMission3

section *MyMission3App* **parents** *scj_prelude, MissionId, MissionIds,*
SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission3MethChan
, MethodCallBindingChannels

process *MyMission3App* $\hat{=}$ **begin**

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{MyMission3MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH3SID} ! \textit{MyMission3MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission3MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$
 $\left(\begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MyMission3MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission3MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end_mission_app} . \textit{MyMission3MID} \longrightarrow \mathbf{Skip})$

end

5.8 Schedulables of MyMission3

section *MyPEH3App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

process *MyPEH3App* $\hat{=}$
 m : *MissionID* • **begin**

state <i>State</i> <i>count</i> : \mathbb{Z}
--

state *State*

initial <i>Init</i> <i>State</i> '
<i>count</i> = 0

handleAsyncEvent $\hat{=}$

$$\left(\begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH3SID} \longrightarrow \\ \left(\begin{array}{l} \text{count} := \text{count} + 1; \\ \text{if } (\text{count} = 10) \longrightarrow \\ \quad \text{requestTerminationCall} . m . \text{MyPEH3SID} \longrightarrow \\ \quad \text{requestTerminationRet} . m . \text{MyPEH3SID} ? \text{requestTermination} \longrightarrow \\ \quad \text{Skip} \\ \square \neg (\text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{MyPEH3SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Methods*) \triangle (*end_periodic_app* . *MyPEH3SID* \longrightarrow **Skip**)

end