nestedSequencer5

Tight Rope v0.75

12th February 2017

1 ID Files

1.1 MissionIds

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

$$
\begin{array}{l}
TopMissionMID : MissionID \\
MidMissionAMID : MissionID \\
BottomMissionAMID : MissionID \\
MidMissionBMID : MissionID \\
BottomMissionBMID : MissionID
\end{array}
$$

$$
\begin{array}{l}
distinct\langle nullMissionId, TopMissionMID, MidMissionAMID, \\
BottomMissionAMID, MidMissionBMID, \\
BottomMissionBMID\rangle
\end{array}
$$

1.2 SchedulablesIds

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

$$
\begin{array}{l}
TopSequencerSID : SchedulableID \\
MT1SID : SchedulableID \\
MidMissionSequencerSID : SchedulableID \\
BottomMissionSequencerASID : SchedulableID \\
OSEHSID : SchedulableID \\
MT2SID : SchedulableID \\
BottomMissionSequencerBSID : SchedulableID \\
APEHSID : SchedulableID \\
PEHSID : SchedulableID \\
\hline
distinct\langle nullSequencerId, nullSchedulableId, TopSequencerSID, \\
MT1SID, MidMissionSequencerSID, \\
BottomMissionSequencerASID, OSEHSID, \\
MT2SID, BottomMissionSequencerBSID, \\
APEHSID, PEHSID\rangle
\end{array}
$$

1.3 Non-Paradigm Objects

1.4 ThreadIds

section *ThreadIds* **parents** *scj_prelude*, *GlobalTypes*

$$SafeletTId : ThreadID$$
$$nullThreadId : ThreadID$$

$$distinct\langle SafeletTId, nullThreadId\rangle$$

1.5 ObjectIds

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

$$\vdash \; distinct\langle\rangle$$

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* . *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 *SafeltAppSync* $\widehat{=}$
{| *getSequencerCall*, *getSequencerRet*, *initializeApplicationCall*, *initializeApplicationRet*, *end_safelet_app* |}

channelset *MissionSequencerAppSync* ==
{| *getNextMissionCall*, *getNextMissionRet*, *end_sequencer_app* |}

channelset *MissionAppSync* ==
{| *initializeCall*, *register*, *initializeRet*, *cleanupMissionCall*, *cleanupMissionRet* |}

channelset *AppSync* ==
$\bigcup$ {*SafeltAppSync*, *MissionSequencerAppSync*, *MissionAppSync*,
MTAppSync, *OSEHSync*, *APEHSync*, *PEHSync*,
{| *getSequencer*, *end_mission_app*, *end_managedThread_app*,
setCeilingPriority, *requestTerminationCall*, *requestTerminationRet*, *terminationPendingCall*,
terminationPendingRet, *handleAsyncEventCall*, *handleAsyncEventRet* |}}

channelset *ThreadSync* ==
{| *raise_thread_priority*, *lower_thread_priority*, *isInterruptedCall*, *isInterruptedRet*, *get_priorityLevel* |}

channelset *LockingSync* ==
{| *lockAcquired*, *startSyncMeth*, *endSyncMeth*, *waitCall*, *waitRet*, *notify*, *isInterruptedCall*, *isInterruptedRet*,
interruptedCall, *interruptedRet*, *done_toplevel_sequencer*, *get_priorityLevel* |}

channelset $Tier0Sync ==$
$\{|\ done\_toplevel\_sequencer, done\_safeletFW,$
$start\_mission\ .\ MidMissionA, done\_mission\ .\ MidMissionA,$
$initializeRet\ .\ MidMissionA, requestTermination\ .\ MidMissionA\ .\ TopSequencer\ |\}$

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 Locking

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

process *Threads* $\widehat{=}$
$(\mathbf{Skip})$

process *Objects* $\widehat{=}$
$(\mathbf{Skip})$

process *Locking* $\widehat{=}$ *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, *TopSequencerApp*, *TopMissionApp*, *MT*1*App*, *MidMissionSequencerApp*
, *MidMissionAApp*, *BottomMissionSequencerAApp*, *BottomMissionAApp*, *MT*2*App*, *OSEHApp*
, *MidMissionBApp*, *BottomMissionSequencerBApp*, *BottomMissionBApp*, *APEHApp*, *PEHApp*

process $ControlTier \mathrel{\widehat=}$
$$\left(\begin{array}{l} SafeletFW \\ \quad [\![ControlTierSync]\!] \\ TopLevelMissionSequencerFW\,(TopSequencer) \end{array}\right)$$

process $Tier0 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW\,(TopMissionID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} ManagedThreadFW\,(MT1ID) \\ \quad [\![SchedulablesSync]\!] \\ SchedulableMissionSequencerFW\,(MidMissionSequencerID) \end{array}\right) \end{array}\right)$$

process $Tier1 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW\,(MidMissionAID) \\ \quad [\![MissionSync]\!] \\ (SchedulableMissionSequencerFW\,(BottomMissionSequencerAID)) \end{array}\right)$$

process $Tier2 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW\,(BottomMissionAID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} ManagedThreadFW\,(MT2ID) \\ \quad [\![SchedulablesSync]\!] \\ OneShotEventHandlerFW\,(OSEHID, (time(60,0)), (time(100,0), nullSchedulableId)) \end{array}\right) \end{array}\right)$$
$$[\![ClusterSync]\!]$$
$$\left(\begin{array}{l} MissionFW\,(MidMissionBID) \\ \quad [\![MissionSync]\!] \\ (SchedulableMissionSequencerFW\,(BottomMissionSequencerBID)) \end{array}\right)$$

process $Tier3 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW\,(BottomMissionBID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} AperiodicEventHandlerFW\,(APEHID, aperiodic, (time(5,0), nullSchedulableId)) \\ \quad [\![SchedulablesSync]\!] \\ PeriodicEventHandlerFW\,(PEHID, (time(60,0), time(5,0), NULL, nullSchedulableId)) \end{array}\right) \end{array}\right)$$

process $Framework \mathrel{\widehat=}$
$$\left(\begin{array}{l} ControlTier \\ \quad [\![TierSync]\!] \\ \left(\begin{array}{l} Tier0 \\ \quad [\![Tier0Sync]\!] \\ Tier1 \\ \quad [\![Tier1Sync]\!] \\ Tier2 \\ \quad [\![Tier2Sync]\!] \\ Tier3 \end{array}\right) \end{array}\right)$$

process $Application \mathrel{\widehat{=}}$

$$
\left(\begin{array}{l}
MyAppApp \\
|||\\
TopSequencerApp \\
|||\\
TopMissionApp \\
|||\\
MT1App \\
|||\\
MidMissionSequencerApp \\
|||\\
MidMissionAApp \\
|||\\
BottomMissionSequencerAApp \\
|||\\
BottomMissionAApp \\
|||\\
MT2App \\
|||\\
OSEHApp(BottomMissionAID) \\
|||\\
MidMissionBApp \\
|||\\
BottomMissionSequencerBApp \\
|||\\
BottomMissionBApp \\
|||\\
APEHApp(BottomMissionBID) \\
|||\\
PEHApp(apehID)
\end{array}\right)
$$

process $Program \mathrel{\widehat{=}} \big(Framework \;[\![\; AppSync \;]\!]\; Application\big) \;[\![\; LockingSync \;]\!]\; Locking$

3 Safelet

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

process $MyAppApp \mathrel{\widehat=} \textbf{begin}$

$$InitializeApplication \mathrel{\widehat=} \begin{pmatrix} initializeApplicationCall \longrightarrow \\ initializeApplicationRet \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$GetSequencer \mathrel{\widehat=} \begin{pmatrix} getSequencerCall \longrightarrow \\ getSequencerRet\,!\,TopSequencerSID \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} GetSequencer \\ \Box \\ InitializeApplication \end{pmatrix} ;\ Methods$$

$\bullet\ (Methods) \triangle (end\_safelet\_app \longrightarrow \textbf{Skip})$

end

4 Top Level Mission Sequencer

section *TopSequencerApp* **parents** *TopLevelMissionSequencerChan*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *TopSequencerClass*, *MethodCallBindingChannels*

process $TopSequencerApp \mathrel{\widehat=}$
$name : String \bullet$ **begin**

State

$this$: **ref** *TopSequencerClass*

state *State*

Init

$State'$

$this' =$ **new** $TopSequencerClass()$

$GetNextMission \mathrel{\widehat=}$ **var** $ret : MissionID \bullet$
$\left(\begin{array}{l} getNextMissionCall \,.\, TopSequencerSID \longrightarrow \\ ret := this \,.\, getNextMission(); \\ getNextMissionRet \,.\, TopSequencerSID \,!\, ret \longrightarrow \\ \textbf{Skip} \end{array}\right)$

$Methods \mathrel{\widehat=}$
$\big(GetNextMission\big) \;;\; Methods$

$\bullet\ (Init \;;\; Methods) \triangle (end\_sequencer\_app \,.\, TopSequencerSID \longrightarrow \textbf{Skip})$

end

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

class *TopSequencerClass* $\widehat{=}$ **begin**

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

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

$$\left(\begin{array}{l} \textbf{if } notReleased = \textbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} this\,.\,notReleased := \textbf{False}; \\ ret := TopMissionMID \end{array}\right) \\ [\!]\ \neg\, notReleased = \textbf{True} \longrightarrow \\ \quad \left(ret := nullMissionId\right) \\ \textbf{fi} \end{array}\right)$$

• **Skip**

end

5 Missions

5.1 TopMission

section *TopMissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *TopMissionMethChan*, *MethodCallBindingChannels*

process $TopMissionApp \mathrel{\widehat=} \textbf{begin}$

State

$this : \textbf{ref}\ TopMissionClass$

state *State*

Init

$State'$

$this' = \textbf{new}\ TopMissionClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ TopMissionMID \longrightarrow \\ register\ !\ MT1SID\ !\ TopMissionMID \longrightarrow \\ register\ !\ MidMissionSequencerSID\ !\ TopMissionMID \longrightarrow \\ initializeRet\ .\ TopMissionMID \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \textbf{var}\ \mathbb{B} : ret \bullet cleanupMissionCall\ .\ TopMissionMID \longrightarrow \\ cleanupMissionRet\ .\ TopMissionMID\ !\ \textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ TopMissionMID \longrightarrow \textbf{Skip})$$

end

5.2 Schedulables of TopMission

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

process $MT1App \mathrel{\widehat=}$ **begin**

$$Run \mathrel{\widehat=} \left(\begin{array}{l} runCall\,.\,MT1SID \longrightarrow \\ \} \\ runRet\,.\,MT1SID \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

$$Methods \mathrel{\widehat=} (Run)\ ;\ Methods$$

$\bullet\ (Methods) \triangle (end\_managedThread\_app\,.\,MT1SID \longrightarrow \textbf{Skip})$

end

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

process $MidMissionSequencerApp \mathrel{\widehat=}$
$name : String \bullet$ **begin**

State

$this$: **ref** *MidMissionSequencerClass*

state *State*

Init

$State'$

$this' =$ **new** $MidMissionSequencerClass()$

$GetNextMission \mathrel{\widehat=}$ **var** $ret : MissionID \bullet$
$\left(\begin{array}{l} getNextMissionCall \,.\, MidMissionSequencerSID \longrightarrow \\ ret := this \,.\, getNextMission(); \\ getNextMissionRet \,.\, MidMissionSequencerSID \,!\, ret \longrightarrow \\ \textbf{Skip} \end{array}\right)$

$Methods \mathrel{\widehat=}$
$\big(GetNextMission\big)\ ;\ Methods$

$\bullet\ (Init\ ;\ Methods) \triangle (end\_sequencer\_app \,.\, MidMissionSequencerSID \longrightarrow \textbf{Skip})$

end

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

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

state *State*

$$releases : \mathbb{Z}$$

state *State*

initial *Init*

$$State'$$

$$releases' = 0$$

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

$$\left(\begin{array}{l}
\textbf{if } (releases = 0) \longrightarrow \\
\quad \left(\begin{array}{l} releases := releases + 1; \\ ret := MidMissionAMID \end{array}\right) \\
[]\ \neg\ (releases = 0) \longrightarrow \\
\quad \textbf{if } (releases = 1) \longrightarrow \\
\quad \left(\begin{array}{l} releases := releases + 1; \\ ret := MidMissionBMID \end{array}\right) \\
[]\ \neg\ (releases = 1) \longrightarrow \\
\quad \left(ret := nullMissionId\right) \\
\textbf{fi} \\
\textbf{fi}
\end{array}\right)$$

• **Skip**

end

5.3 MidMissionA

section *MidMissionAApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MidMissionAMethChan*, *MethodCallBindingChannels*

process $MidMissionAApp \mathrel{\widehat=}$ **begin**

State

$this : \mathbf{ref}\ MidMissionAClass$

state *State*

Init

$State'$

$this' = \mathbf{new}\ MidMissionAClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ MidMissionAMID \longrightarrow \\ register\ !\ BottomMissionSequencerASID\ !\ MidMissionAMID \longrightarrow \\ initializeRet\ .\ MidMissionAMID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \mathbf{var}\ \mathbb{B} : ret \bullet cleanupMissionCall\ .\ MidMissionAMID \longrightarrow \\ cleanupMissionRet\ .\ MidMissionAMID\ !\ \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ MidMissionAMID \longrightarrow \mathbf{Skip})$

end

5.4 Schedulables of MidMissionA

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

process $BottomMissionSequencerAApp \mathrel{\widehat=}$
$name : String \bullet$ **begin**

State

$this$: **ref** *BottomMissionSequencerAClass*

state *State*

Init

$State'$

$this' =$ **new** $BottomMissionSequencerAClass()$

$GetNextMission \mathrel{\widehat=}$ **var** $ret : MissionID \bullet$
$\left(\begin{array}{l} getNextMissionCall\,.\,BottomMissionSequencerASID \longrightarrow \\ ret := this\,.\,getNextMission(); \\ getNextMissionRet\,.\,BottomMissionSequencerASID\,!\,ret \longrightarrow \\ \textbf{Skip} \end{array}\right)$

$Methods \mathrel{\widehat=}$
$\big(GetNextMission\big)\ ;\ Methods$

$\bullet\ (Init\ ;\ Methods) \triangle (end\_sequencer\_app\,.\,BottomMissionSequencerASID \longrightarrow \textbf{Skip})$

end

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

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

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

$$\left(\begin{array}{l} \textbf{if } notReleased = \textbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} this\,.\,notReleased := \textbf{False}; \\ ret := BottomMissionAMID \end{array}\right) \\ [\!]\ \neg\, notReleased = \textbf{True} \longrightarrow \\ \quad \left(ret := nullMissionId\right) \\ \textbf{fi} \end{array}\right)$$

• **Skip**

end

5.5 BottomMissionA

section *BottomMissionAApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *BottomMissionAMethChan*, *MethodCallBindingChannels*

process $BottomMissionAApp \mathrel{\widehat=}$ **begin**

State

$this : \mathbf{ref}\ BottomMissionAClass$

state *State*

Init

$State'$

$this' = \mathbf{new}\ BottomMissionAClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ BottomMissionAMID \longrightarrow \\ register\ !\ OSEHSID\ !\ BottomMissionAMID \longrightarrow \\ register\ !\ MT2SID\ !\ BottomMissionAMID \longrightarrow \\ initializeRet\ .\ BottomMissionAMID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \mathbf{var}\ \mathbb{B} : ret \bullet cleanupMissionCall\ .\ BottomMissionAMID \longrightarrow \\ cleanupMissionRet\ .\ BottomMissionAMID\ !\ \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ BottomMissionAMID \longrightarrow \mathbf{Skip})$$

end

5.6 Schedulables of BottomMissionA

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

process $MT2App \mathrel{\widehat=}$ **begin**

$$Run \mathrel{\widehat=} \left(\begin{array}{l} runCall\,.\,MT2SID \longrightarrow \\ \} \\ runRet\,.\,MT2SID \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

$$Methods \mathrel{\widehat=} (Run)\ ;\ Methods$$

- $(Methods) \triangle (end\_managedThread\_app\,.\,MT2SID \longrightarrow \textbf{Skip})$

end

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

process $OSEHApp \mathrel{\widehat=}$
$controllingMission : MissionID \bullet$ **begin**

State

$this : OSEHClass$

state *State*

Init

$State'$

$this' = \textbf{new}\ OSEHClass()$

$$handleAsyncEvent \mathrel{\widehat=} \left(\begin{array}{l} handleAsyncEventCall\,.\,OSEHSID \longrightarrow \\ \left(\begin{array}{l} requestTerminationCall\,.\,controllingMision\,.\,OSEHSID \longrightarrow \\ requestTerminationRet\,.\,controllingMision\,.\,OSEHSID\,?\,requestTermination \longrightarrow \\ \textbf{Skip} \end{array} \right) ; \\ handleAsyncEventRet\,.\,OSEHSID \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

$$Methods \mathrel{\widehat=} \big(handleAsyncEvent\big)\ ;\ Methods$$

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_oneShot\_app\,.\,OSEHSID \longrightarrow \textbf{Skip})$$

end

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

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

state *State*

$controllingMision : Mission$

state *State*

initial *Init*

$State'$

- **Skip**

end

5.7 MidMissionB

section *MidMissionBApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MidMissionBMethChan*, *MethodCallBindingChannels*

process $MidMissionBApp \mathrel{\widehat=}$ **begin**

State

$this : \mathbf{ref}\ MidMissionBClass$

state *State*

Init

$State'$

$this' = \mathbf{new}\ MidMissionBClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ MidMissionBMID \longrightarrow \\ register\ !\ BottomMissionSequencerBSID\ !\ MidMissionBMID \longrightarrow \\ initializeRet\ .\ MidMissionBMID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \mathbf{var}\ \mathbb{B} : ret \bullet cleanupMissionCall\ .\ MidMissionBMID \longrightarrow \\ cleanupMissionRet\ .\ MidMissionBMID\ !\ \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ MidMissionBMID \longrightarrow \mathbf{Skip})$

end

5.8 Schedulables of MidMissionB

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

process $BottomMissionSequencerBApp \mathrel{\widehat=}$
$name : String \bullet$ **begin**

State

$this$: **ref** *BottomMissionSequencerBClass*

state *State*

Init

$State'$

$this' =$ **new** $BottomMissionSequencerBClass()$

$GetNextMission \mathrel{\widehat=}$ **var** $ret : MissionID \bullet$
$\left(\begin{array}{l} getNextMissionCall\,.\,BottomMissionSequencerBSID \longrightarrow \\ ret := this\,.\,getNextMission(); \\ getNextMissionRet\,.\,BottomMissionSequencerBSID\,!\,ret \longrightarrow \\ \textbf{Skip} \end{array}\right)$

$Methods \mathrel{\widehat=}$
$\big(GetNextMission\big)\ ;\ Methods$

$\bullet\ (Init\ ;\ Methods) \triangle (end\_sequencer\_app\,.\,BottomMissionSequencerBSID \longrightarrow \textbf{Skip})$

end

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

class *BottomMissionSequencerBClass* $\widehat{=}$ **begin**

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

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

$$\left(\begin{array}{l} \textbf{if } notReleased = \textbf{True} \longrightarrow \\ \quad \left(\begin{array}{l} this\,.\,notReleased := \textbf{False}; \\ ret := BottomMissionBMID \end{array}\right) \\ [\!]\ \neg\, notReleased = \textbf{True} \longrightarrow \\ \quad \left(ret := nullMissionId\right) \\ \textbf{fi} \end{array}\right)$$

• **Skip**

end

5.9 BottomMissionB

section *BottomMissionBApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *BottomMissionBMethChan*, *MethodCallBindingChannels*

process $BottomMissionBApp \mathrel{\widehat=}$ **begin**

State

$this : \mathbf{ref}\ BottomMissionBClass$

state *State*

Init

$State'$

$this' = \mathbf{new}\ BottomMissionBClass()$

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ BottomMissionBMID \longrightarrow \\ register\ !\ APEHSID\ !\ BottomMissionBMID \longrightarrow \\ register\ !\ PEHSID\ !\ BottomMissionBMID \longrightarrow \\ initializeRet\ .\ BottomMissionBMID \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} \mathbf{var}\ \mathbb{B} : ret \bullet cleanupMissionCall\ .\ BottomMissionBMID \longrightarrow \\ cleanupMissionRet\ .\ BottomMissionBMID\ !\ \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$Methods \mathrel{\widehat=} \begin{pmatrix} InitializePhase \\ \Box \\ CleanupPhase \end{pmatrix} ;\ Methods$$

$$\bullet\ (Init\ ;\ Methods) \triangle (end\_mission\_app\ .\ BottomMissionBMID \longrightarrow \mathbf{Skip})$$

end

5.10 Schedulables of BottomMissionB

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

process $APEHApp \mathrel{\widehat=}$
$controllingMission : MissionID \bullet$ **begin**

$handleAsyncEvent \mathrel{\widehat=}$
$\left(\begin{array}{l} handleAsyncEventCall\,.\,APEHSID \longrightarrow \\ \left(\begin{array}{l} requestTerminationCall\,.\,controllingMission\,.\,APEHSID \longrightarrow \\ requestTerminationRet\,.\,controllingMission\,.\,APEHSID\,?\,requestTermination \longrightarrow \\ \mathbf{Skip} \end{array}\right) ; \\ handleAsyncEventRet\,.\,APEHSID \longrightarrow \\ \mathbf{Skip} \end{array}\right)$

$Methods \mathrel{\widehat=}$
$\big(handleAsyncEvent\big)\ ;\ Methods$

$\bullet\ (Methods) \triangle (end\_aperiodic\_app\,.\,APEHSID \longrightarrow \mathbf{Skip})$

end

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

process $PEHApp \mathrel{\widehat=}$
$\quad apeh : SchedulableID \bullet \textbf{begin}$

$$handleAsyncEvent \mathrel{\widehat=} \left(\begin{array}{l} handleAsyncEventCall\,.\,PEHSID \longrightarrow \\ \left(\begin{array}{l} releaseCall\,.\,apeh\,.\,PEHSID \longrightarrow \\ \textbf{Skip} \end{array}\right); \\ handleAsyncEventRet\,.\,PEHSID \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

$$Methods \mathrel{\widehat=} \big(handleAsyncEvent\big)\ ;\ \ Methods$$

$$\bullet\ (Methods) \triangle (end\_periodic\_app\,.\,PEHSID \longrightarrow \textbf{Skip})$$

end