

nestedSequencer4

Tight Rope v0.75

1st March 2017

1 ID Files

1.1 MissionIds

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

TopMissionMID : *MissionID*
MidMissionMID : *MissionID*
BottomMissionAMID : *MissionID*
BottomMissionBMID : *MissionID*

distinct(*nullMissionId*, *TopMissionMID*, *MidMissionMID*,
BottomMissionAMID, *BottomMissionBMID*)

1.2 SchedulablesIds

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

TopSequencerSID : SchedulableID

MT1SID : SchedulableID

MidMissionSequencerSID : SchedulableID

BottomMissionSequencerASID : SchedulableID

BottomMissionSequencerBSID : SchedulableID

MT2SID : SchedulableID

MT3SID : SchedulableID

*distinct (nullSequencerId, nullSchedulableId, TopSequencerSID,
MT1SID, MidMissionSequencerSID,
BottomMissionSequencerASID, BottomMissionSequencerBSID,
MT2SID, MT3SID)*

1.3 Non-Paradigm Objects

1.4 ThreadIds

section *ThreadId* **parents** *scj_prelude, GlobalTypes*

SafeletTid : *ThreadID*
nullThreadId : *ThreadID*

distinct(*SafeletTid*, *nullThreadId*)

1.5 ObjectIds

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

$\text{distinct} \langle \rangle$

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* $\hat{=}$
{ *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 . 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 Locking

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

process *Threads* $\hat{=}$
(**Skip**)

process *Objects* $\hat{=}$
(**Skip**)

process *Locking* $\hat{=}$ (*Threads* \llbracket *ThreadSync* \rrbracket *Objects*) \triangle (*done_toplevel_sequencer* \longrightarrow **Skip**)

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, 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* $\hat{=}$

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

process *Tier1* $\hat{=}$

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

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* $\hat{=}$

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

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

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

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

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

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

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

state *State*

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

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

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

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

end

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

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

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

state *State*

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

protected *getNextMission* $\hat{=}$

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

• **Skip**

end

5 Missions

5.1 TopMission

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

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

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

state *State*

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

InitializePhase $\hat{=}$

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

CleanupPhase $\hat{=}$

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

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

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

end

5.2 Schedulables of TopMission

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

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

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

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

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

end

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

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

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

state *State*

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

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

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

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

end

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

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

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

state *State*

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

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } notReleased \longrightarrow \\ \quad \left(notReleased := \mathbf{False}; \right. \\ \quad \left. ret := MidMissionMID \right) \\ \parallel \neg notReleased \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5.3 MidMission

section *MidMissionApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MidMissionMethChan*,
MethodCallBindingChannels

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

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

state *State*

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

InitializePhase $\hat{=}$

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

CleanupPhase $\hat{=}$

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

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

• (*Init* ; *Methods*) \triangle (*end_mission_app* . *MidMissionMID* \longrightarrow **Skip**)

end

5.4 Schedulables of MidMission

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

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

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

state *State*

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

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

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

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

end

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

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

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

state *State*

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

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } notReleased \longrightarrow \\ \quad \left(notReleased := \mathbf{False}; \right. \\ \quad \left. ret := BottomMissionAMID \right) \\ \parallel \neg notReleased \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

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

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

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

state *State*

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

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

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

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

end

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

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

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

state *State*

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

protected *getNextMission* $\hat{=}$

$$\left(\begin{array}{l} \text{if } notReleased \longrightarrow \\ \quad \left(notReleased := \mathbf{False}; \right. \\ \quad \left. ret := BottomMissionBMID \right) \\ \parallel \neg notReleased \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

end

5.5 BottomMissionA

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

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

State

this : **ref** *BottomMissionAClass*

state *State*

Init

State'

this' = **new** *BottomMissionAClass*()

InitializePhase $\hat{=}$

$\left(\begin{array}{l} \text{initializeCall} . \text{BottomMissionAMID} \longrightarrow \\ \text{register} ! \text{MT2SID} ! \text{BottomMissionAMID} \longrightarrow \\ \text{initializeRet} . \text{BottomMissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

CleanupPhase $\hat{=}$

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

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

• (*Init* ; *Methods*) \triangle (*end_mission_app* . *BottomMissionAMID* \longrightarrow **Skip**)

end

5.6 Schedulables of BottomMissionA

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

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

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

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

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

end

5.7 BottomMissionB

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

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

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

state *State*

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

InitializePhase $\hat{=}$
 $\left(\begin{array}{l} \textit{initializeCall} . \textit{BottomMissionBMID} \longrightarrow \\ \textit{register} ! \textit{MT3SID} ! \textit{BottomMissionBMID} \longrightarrow \\ \textit{initializeRet} . \textit{BottomMissionBMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

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

end

5.8 Schedulables of BottomMissionB

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

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

Run $\hat{=}$
$$\left(\begin{array}{l} \text{runCall} . \text{MT3SID} \longrightarrow \\ (\mathbf{Skip}) ; \\ \text{runRet} . \text{MT3SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

• $(\text{Methods}) \triangle (\text{end_managedThread_app} . \text{MT3SID} \longrightarrow \mathbf{Skip})$

end