nestedSequencer4

Tight Rope v0.88

4th March 2017

1 ID Files

1.1 MissionIds

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

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

$$
distinct\langle nullMissionId, TopMissionMID, MidMissionMID, BottomMissionAMID, BottomMissionBMID\rangle
$$

1.2 SchedulablesIds

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

$$
\begin{array}{l}
TopSequencerSID : SchedulableID \\
MT1SID : SchedulableID \\
MidMissionSequencerSID : SchedulableID \\
BottomMissionSequencerASID : SchedulableID \\
BottomMissionSequencerBSID : SchedulableID \\
MT2SID : SchedulableID \\
MT3SID : SchedulableID \\
\hline
distinct\langle nullSequencerId, nullSchedulableId, TopSequencerSID, \\
MT1SID, MidMissionSequencerSID, \\
BottomMissionSequencerASID, BottomMissionSequencerBSID, \\
MT2SID, MT3SID\rangle
\end{array}
$$

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

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

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

process *Locking* $\widehat{=}$ $(Threads \llbracket ThreadSync \rrbracket Objects) \triangle (done\_toplevel\_sequencer \longrightarrow \mathbf{Skip})$

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*, *MT*1*App*, *MidMissionSequencerApp*
, *MidMissionApp*, *BottomMissionSequencerAApp*, *BottomMissionSequencerBApp*
, *BottomMissionAApp*, *MT*2*App*, *BottomMissionBApp*, *MT*3*App*

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

process $Tier2 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(BottomMissionAID) \\ \quad [\![MissionSync]\!] \\ (ManagedThreadFW(MT2ID)) \end{array}\right)$$

process $Tier3 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(BottomMissionBID) \\ \quad [\![MissionSync]\!] \\ (ManagedThreadFW(MT3ID)) \end{array}\right)$$

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

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

process $Program \mathrel{\widehat{=}} \big(Framework \llbracket\ AppSync\ \rrbracket\ Application\big) \llbracket\ LockingSync\ \rrbracket\ Locking$

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 $TopSequencerApp \mathrel{\widehat=}$ **begin**

State

$this : \mathbf{ref}\ TopSequencerClass$

state *State*

Init

$State'$

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

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

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

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

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* $\widehat{=}$

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

- **Skip**

end

5 Missions

5.1 TopMission

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

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

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

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\,.\,TopMissionMID \longrightarrow \\ cleanupMissionRet\,.\,TopMissionMID\,!\,\textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

5.2 Schedulables of TopMission

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

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

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

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

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

end

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

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

State

$this : \textbf{ref}\ MidMissionSequencerClass$

state *State*

Init

$State'$

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

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

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

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

end

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

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased' = true$$

protected *getNextMission* $\widehat{=}$

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

- **Skip**

end

5.3 MidMission

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

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,MidMissionMID \longrightarrow \\ register\,!\,BottomMissionSequencerASID\,!\,MidMissionMID \longrightarrow \\ register\,!\,BottomMissionSequencerBSID\,!\,MidMissionMID \longrightarrow \\ initializeRet\,.\,MidMissionMID \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\,.\,MidMissionMID \longrightarrow \\ cleanupMissionRet\,.\,MidMissionMID\,!\,\textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

5.4 Schedulables of MidMission

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

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

State

$this : \textbf{ref}\ BottomMissionSequencerAClass$

state *State*

Init

$State'$

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

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

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

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

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* $\widehat{=}$

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

- **Skip**

end

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

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

State

$$this : \textbf{ref}\ BottomMissionSequencerBClass$$

state *State*

Init

$$State'$$

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

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

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

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

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* $\widehat{=}$

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

- **Skip**

end

5.5 BottomMissionA

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

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

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

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\,.\,BottomMissionAMID \longrightarrow \\ cleanupMissionRet\,.\,BottomMissionAMID\,!\,\mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

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

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

end

5.6 Schedulables of BottomMissionA

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

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

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

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

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

end

5.7 BottomMissionB

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

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\,.\,BottomMissionBMID \longrightarrow \\ register\,!\,MT3SID\,!\,BottomMissionBMID \longrightarrow \\ initializeRet\,.\,BottomMissionBMID \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

$$CleanupPhase \mathrel{\widehat=} \begin{pmatrix} cleanupMissionCall\,.\,BottomMissionBMID \longrightarrow \\ cleanupMissionRet\,.\,BottomMissionBMID\,!\,\textbf{True} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

5.8 Schedulables of BottomMissionB

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

process $MT3App \mathrel{\widehat=}$ **begin**

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

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

- $(Methods) \triangle (end\_managedThread\_app\,.\,MT3SID \longrightarrow \mathbf{Skip})$

end