Flatbuffer

Tight Rope v0.65

12th February 2016

1 ID Files

1.1 MissionIds

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

$$\mathit{FlatBufferMissionMID} : \mathit{MissionID}$$

$$\mathit{distinct}\langle \mathit{nullMissionId}, \mathit{FlatBufferMissionMID}\rangle$$

1.2 SchedulablesIds

$\mathbf{section}\ SchedulableIds\ \mathbf{parents}\ scj\_prelude, SchedulableId$

$$
\begin{array}{|l}
FlatBufferMissionSequencerSID : SchedulableID \\
ReaderSID : SchedulableID \\
WriterSID : SchedulableID \\
\hline
distinct \langle nullSequencerId, nullSchedulableId, FlatBufferMissionSequencerSID, \\
ReaderSID, WriterSID \rangle
\end{array}
$$

1.3 ThreadIds

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

$$
\begin{array}{|l}
WriterTID : ThreadID \\
ReaderTID : ThreadID \\
\hline
distinct \langle SafeletTID, nullTID, \\
WriterTID, ReaderTID \rangle
\end{array}
$$

1.4 ObjectIds

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

$$\begin{array}{|l}
\mathit{FlatBufferMissionOID} : \mathit{ObjectID} \\
\hline
\mathit{distinct}\langle \mathit{FlatBufferMissionOID} \rangle
\end{array}$$

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* . *FlatBufferMission*, *done_mission* . *FlatBufferMission*,
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* ⦄

2.2 MethodCallBinder

vv

$\mathbf{section}\ \mathit{MethodCallBindingChannels}\ \mathbf{parents}\ \mathit{scj\_prelude}, \mathit{GlobalTypes}, \mathit{MissionId}, \mathit{SchedulableId}, \mathit{ThreadId}$

$\mathbf{channel}\ \mathit{binder\_readCall} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadId}$
$\mathbf{channel}\ \mathit{binder\_readRet} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadId} \times \mathbb{Z}$

$\mathit{readLocs} == \{\mathit{FlatBufferMissionMID}\}$
$\mathit{readCallers} == \{\mathit{ReaderSID}\}$

$\mathbf{channel}\ \mathit{binder\_writeCall} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadId} \times \mathbb{Z}$
$\mathbf{channel}\ \mathit{binder\_writeRet} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadId}$

$\mathit{writeLocs} == \{\mathit{FlatBufferMissionMID}\}$
$\mathit{writeCallers} == \{\mathit{WriterSID}\}$

$\mathbf{channelset}\ \mathit{MethodCallBinderSync} == \{\!|\ \mathit{done\_toplevel\_sequencer},$
$\mathit{binder\_readCall}, \mathit{binder\_readRet},$
$\mathit{binder\_writeCall}, \mathit{binder\_writeRet}\ |\!\}$

$\mathbf{process}\ \mathit{MethodCallBinder} \mathrel{\widehat=} \mathbf{begin}$

$$\mathit{read\_MethodBinder} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{binder\_readCall} \\ \qquad ?\, \mathit{loc} : (\mathit{loc} \in \mathit{readLocs}) \\ \qquad ?\, \mathit{caller} : (\mathit{caller} \in \mathit{readCallers}) \\ \qquad ?\, \mathit{callingThread} \longrightarrow \\ \mathit{readCall} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread} \longrightarrow \\ \mathit{readRet} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread}\ ?\, \mathit{ret} \longrightarrow \\ \mathit{binder\_readRet} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread}\ !\, \mathit{ret} \longrightarrow \\ \mathit{read\_MethodBinder} \end{array}\right)$$

$$\mathit{write\_MethodBinder} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{binder\_writeCall} \\ \qquad ?\, \mathit{loc} : (\mathit{loc} \in \mathit{writeLocs}) \\ \qquad ?\, \mathit{caller} : (\mathit{caller} \in \mathit{writeCallers}) \times \mathbb{Z} \\ \qquad ?\, \mathit{callingThread} \longrightarrow \\ \mathit{writeCall} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread} \times \mathbb{Z} \longrightarrow \\ \mathit{writeRet} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread} \longrightarrow \\ \mathit{binder\_writeRet} \,.\, \mathit{loc} \,.\, \mathit{caller} \,.\, \mathit{callingThread} \longrightarrow \\ \mathit{write\_MethodBinder} \end{array}\right)$$

$$\mathit{BinderActions} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{read\_MethodBinder} \\ ||| \\ \mathit{write\_MethodBinder} \end{array}\right)$$

$\bullet\ \mathit{BinderActions} \mathbin{\triangle} (\mathit{done\_toplevel\_sequencer} \longrightarrow \mathbf{Skip})$

$\mathbf{end}$

$\mathbf{process}\ \mathit{ApplicationB} \mathrel{\widehat=} \mathit{Application} \llbracket\ \mathit{MethodCallBinderSync}\ \rrbracket\ \mathit{MethodCallBinder}$

2.3 Locking

process *Threads* $\widehat{=}$

$$\left(\begin{array}{l} \mathit{ThreadFW}(\mathit{WriterTID}, 10) \\ ||| \\ \mathit{ThreadFW}(\mathit{ReaderTID}, 10) \end{array}\right)$$

process *Objects* $\widehat{=}$

$\left(\mathit{ObjectFW}(\mathit{FlatBufferMissionOID})\right)$

process *Locking* $\widehat{=}$ *Threads* $[\![$ *ThreadSync* $]\!]$ *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*,
FlatBufferApp, *FlatBufferMissionSequencerApp*, *FlatBufferMissionApp*, *ReaderApp*, *WriterApp*

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

$$
\begin{array}{l}
\mathbf{process}\ \mathit{Tier0} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{MissionFW}\,(\mathit{FlatBufferMissionID}) \\
\qquad [\![\mathit{MissionSync}]\!] \\
\left(\begin{array}{l}
\mathit{ManagedThreadFW}\,(\mathit{ReaderID}) \\
\qquad [\![\mathit{SchedulablesSync}]\!] \\
\mathit{ManagedThreadFW}\,(\mathit{WriterID})
\end{array}\right)
\end{array}\right)
\end{array}
$$

$$
\begin{array}{l}
\mathbf{process}\ \mathit{Framework} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{ControlTier} \\
\qquad [\![\mathit{TierSync}]\!] \\
(\,\mathit{Tier0})
\end{array}\right)
\end{array}
$$

$$
\begin{array}{l}
\mathbf{process}\ \mathit{Application} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{FlatBufferApp} \\
\interleave \\
\mathit{FlatBufferMissionSequencerApp} \\
\interleave \\
\mathit{FlatBufferMissionApp} \\
\interleave \\
\mathit{ReaderApp}(\mathit{FlatBufferMissionID}) \\
\interleave \\
\mathit{WriterApp}(\mathit{FlatBufferMissionID})
\end{array}\right)
\end{array}
$$

$$
\mathbf{process}\ \mathit{Program} \mathrel{\widehat=} \big(\mathit{Framework}\ [\![\ \mathit{AppSync}\ ]\!]\ \mathit{ApplicationB}\big)\ [\![\ \mathit{LockingSync}\ ]\!]\ \mathit{Locking}
$$

3 Safelet

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

$\textbf{process}\ \mathit{FlatBufferApp} \mathrel{\widehat=} \textbf{begin}$

$$\mathit{InitializeApplication} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{initializeApplicationCall} \longrightarrow \\ \mathit{initializeApplicationRet} \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

$$\mathit{GetSequencer} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{getSequencerCall} \longrightarrow \\ \mathit{getSequencerRet}\,!\,\mathit{FlatBufferMissionSequencerSID} \longrightarrow \\ \textbf{Skip} \end{array}\right)$$

$$\mathit{Methods} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{GetSequencer} \\ \Box \\ \mathit{InitializeApplication} \end{array}\right) \mathrel{;}\ \mathit{Methods}$$

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

end

4 Top Level Mission Sequencer

$\mathbf{section}\ \mathit{FlatBufferMissionSequencerApp}\ \mathbf{parents}\ \mathit{TopLevelMissionSequencerChan},$
$\quad \mathit{MissionId}, \mathit{MissionIds}, \mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{MethodCallBindingChannels}, \mathit{FlatBufferMissionSequencerClass}$

$\mathbf{process}\ \mathit{FlatBufferMissionSequencerApp} \mathrel{\widehat=} \mathbf{begin}$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{State}} \\
\hline
\mathit{this} : \mathbf{ref}\ \mathit{FlatBufferMissionSequencerClass} \\
\hline
\end{array}
$$

$\mathbf{state}\ \mathit{State}$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Init}} \\
\hline
\mathit{State}\,' \\
\hline
\mathit{this}' = \mathbf{new}\ \mathit{FlatBufferMissionSequencerClass}() \\
\hline
\end{array}
$$

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

$$
\begin{array}{l}
\mathit{Methods} \mathrel{\widehat=} \\
\left(\mathit{GetNextMission}\right) \mathrel{;}\ \mathit{Methods}
\end{array}
$$

$\bullet\ (\mathit{Init} \mathrel{;}\ \mathit{Methods}) \mathbin{\triangle} (\mathit{end\_sequencer\_app} \,.\, \mathit{FlatBufferMissionSequencerSID} \longrightarrow \mathbf{Skip})$

$\mathbf{end}$

$\mathbf{section}\ \mathit{FlatBufferMissionSequencerClass}\ \mathbf{parents}\ \mathit{scj\_prelude}, \mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{SafeletChan}, \mathit{MethodCallB}$
$, \mathit{MissionId}, \mathit{MissionIds}$

$\mathbf{class}\ \mathit{FlatBufferMissionSequencerClass} \mathrel{\widehat=} \mathbf{begin}$

$\mathbf{state}\ \mathit{State}$

$\mathit{returnedMission} : \mathbb{B}$

$\mathbf{state}\ \mathit{State}$

$\mathbf{initial}\ \mathit{Init}$

$\mathit{State}\,'$

$\mathit{returnedMission}' = \mathbf{False}$

$$\begin{array}{l}
\mathbf{protected}\ \mathit{getNextMission} \mathrel{\widehat=} \mathbf{var}\ \mathit{ret} : \mathit{MissionID} \bullet \\
\left(\begin{array}{l}
\mathbf{if}\ (\neg\ \mathit{returnedMission} = \mathbf{True}) \longrightarrow \\
\qquad \left(\begin{array}{l} \mathit{this} \,.\, \mathit{returnedMission} := \mathbf{True}; \\ \mathit{ret} := \mathit{FlatBufferMissionMID} \end{array}\right) \\
[\!]\ \neg\ (\neg\ \mathit{returnedMission} = \mathbf{True}) \longrightarrow \\
\qquad \left(\mathit{ret} := \mathit{nullMissionId}\right) \\
\mathbf{fi}
\end{array}\right)
\end{array}$$

$\bullet\ \mathbf{Skip}$

$\mathbf{end}$

5 Missions

5.1 FlatBufferMission

section $\mathit{FlatBufferMissionApp}$ **parents** $\mathit{scj\_prelude}, \mathit{MissionId}, \mathit{MissionIds},$
$\mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{MissionChan}, \mathit{SchedulableMethChan}, \mathit{MethodCallBindingChannels}, \mathit{FlatBufferMissionC}$
$, \mathit{ObjectChan}, \mathit{ObjectIds}, \mathit{ThreadIds}, \mathit{FlatBufferMissionMethChan}$

process $\mathit{FlatBufferMissionApp} \mathrel{\widehat=}$ **begin**

State

$\mathit{this} : \mathbf{ref}\ \mathit{FlatBufferMissionClass}$

state State

Init

$\mathit{State}\,'$

$\mathit{this}' = \mathbf{new}\ \mathit{FlatBufferMissionClass}()$

$$
\mathit{InitializePhase} \mathrel{\widehat=}
\left(\begin{array}{l}
\mathit{initializeCall} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{register} \,!\, \mathit{ReaderSID} \,!\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{register} \,!\, \mathit{WriterSID} \,!\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{initializeRet} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
$$

$$
\mathit{CleanupPhase} \mathrel{\widehat=}
\left(\begin{array}{l}
\mathit{cleanupMissionCall} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{cleanupMissionRet} \,.\, \mathit{FlatBufferMissionMID} \,!\, \mathbf{True} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
$$

$$
\mathit{bufferEmptyMeth} \mathrel{\widehat=} \mathbf{var}\ \mathit{ret} : \mathbb{B} \bullet
\left(\begin{array}{l}
\mathit{bufferEmptyCall} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{ret} := \mathit{this} \,.\, \mathit{bufferEmpty}(); \\
\mathit{bufferEmptyRet} \,.\, \mathit{FlatBufferMissionMID} \,!\, \mathit{ret} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
$$

$$
\mathit{cleanUpMeth} \mathrel{\widehat=} \mathbf{var}\ \mathit{ret} : \mathbb{B} \bullet
\left(\begin{array}{l}
\mathit{cleanUpCall} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \\
\mathit{ret} := \mathit{this} \,.\, \mathit{cleanUp}(); \\
\mathit{cleanUpRet} \,.\, \mathit{FlatBufferMissionMID} \,!\, \mathit{ret} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
$$

$$
\begin{array}{l}
\mathit{writeSyncMeth} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{writeCall} \,.\, \mathit{FlatBufferMissionMID} \,?\, \mathit{caller} \,?\, \mathit{thread} \,?\, \mathit{update} \longrightarrow \\
\left(\begin{array}{l}
\mathit{startSyncMeth} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{lockAcquired} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\left(\begin{array}{l}
\left(\begin{array}{l}
\mu X \bullet \\
\left(\begin{array}{l}
\mathbf{var}\ \mathit{loopVar} : \mathbb{B} \bullet \mathit{loopVar} := (\lnot\ \mathit{bufferEmpty}()); \\
\mathbf{if}\ (\mathit{loopVar} = \mathbf{True}) \longrightarrow \\
\qquad \left(\begin{array}{l}
\mathit{waitCall} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{waitRet} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathbf{Skip}
\end{array}\right) ;\ X \\
[\!]\ (\mathit{loopVar} = \mathbf{False}) \longrightarrow \mathbf{Skip} \\
\mathbf{fi}
\end{array}\right)
\end{array}\right) \\
; \\
\mathit{this} \,.\, \mathit{buffer} := \mathit{update}; \\
\mathit{notify} \,.\, \mathit{FlatBufferMissionOID} \,!\, \mathit{thread} \longrightarrow \\
\mathbf{Skip}
\end{array}\right) ; \\
\mathit{endSyncMeth} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{writeRet} \,.\, \mathit{FlatBufferMissionMID} \,.\, \mathit{caller} \,.\, \mathit{thread} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
\end{array}\right)
\end{array}
$$

$$
\begin{array}{l}
\mathit{readSyncMeth} \mathrel{\widehat=} \mathbf{var}\ \mathit{ret} : \mathbb{Z} \bullet \\
\left(\begin{array}{l}
\mathit{readCall} \,.\, \mathit{FlatBufferMissionMID} \,?\, \mathit{caller} \,?\, \mathit{thread} \longrightarrow \\
\left(\begin{array}{l}
\mathit{startSyncMeth} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{lockAcquired} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\left(\begin{array}{l}
\left(\begin{array}{l}
\mu X \bullet \\
\left(\begin{array}{l}
\mathbf{var}\ \mathit{loopVar} : \mathbb{B} \bullet \mathit{loopVar} := \ \mathit{bufferEmpty}(); \\
\mathbf{if}\ (\mathit{loopVar} = \mathbf{True}) \longrightarrow \\
\qquad \left(\begin{array}{l}
\mathit{waitCall} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{waitRet} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathbf{Skip}
\end{array}\right) ;\ X \\
[\!]\ (\mathit{loopVar} = \mathbf{False}) \longrightarrow \mathbf{Skip} \\
\mathbf{fi}
\end{array}\right)
\end{array}\right) \\
; \\
\mathbf{var}\ \mathit{out} : \mathbb{Z} \bullet \mathit{out} := \mathit{this} \,.\, \mathit{buffer}; \\
\mathit{this} \,.\, \mathit{buffer} := 0; \\
\mathit{notify} \,.\, \mathit{FlatBufferMissionOID} \,!\, \mathit{thread} \longrightarrow \\
\mathbf{Skip}; \\
\mathit{ret} := \mathit{out}
\end{array}\right) ; \\
\mathit{endSyncMeth} \,.\, \mathit{FlatBufferMissionOID} \,.\, \mathit{thread} \longrightarrow \\
\mathit{readRet} \,.\, \mathit{FlatBufferMissionMID} \,.\, \mathit{caller} \,.\, \mathit{thread} \,!\, \mathit{ret} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
\end{array}\right)
\end{array}
$$

$$
\mathit{Methods} \mathrel{\widehat=}
\left(\begin{array}{l}
\mathit{InitializePhase} \\
\Box \\
\mathit{CleanupPhase} \\
\Box \\
\mathit{bufferEmptyMeth} \\
\Box \\
\mathit{cleanUpMeth} \\
\Box \\
\mathit{writeSyncMeth} \\
\Box \\
\mathit{readSyncMeth}
\end{array}\right) ;\ \mathit{Methods}
$$

$$
\bullet\ (\mathit{Init}\ ;\ \mathit{Methods}) \mathbin{\triangle} (\mathit{end\_mission\_app} \,.\, \mathit{FlatBufferMissionMID} \longrightarrow \mathbf{Skip})
$$

end

section *FlatBufferMissionClass* **parents** *scj_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingCh*

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

state *State*

$$\begin{array}{|l}
buffer : \mathbb{Z} \\
t : testClass
\end{array}$$

state *State*

initial *Init*

$$\begin{array}{|l}
State~' \\
\hline
buffer' = 0 \\
t' = testClass
\end{array}$$

public *bufferEmpty* $\widehat{=}$ **var** *ret* : $\mathbb{B}$ ●

$$\left(\begin{array}{l}
\mathbf{if}\ (buffer = 0) \longrightarrow \\
\qquad ret := \mathbf{True} \\
[\!]\ \neg\ (buffer = 0) \longrightarrow \\
\qquad ret := \mathbf{False} \\
\mathbf{fi}
\end{array}\right)$$

public *cleanUp* $\widehat{=}$ **var** *ret* : $\mathbb{B}$ ●

$\big(ret := \mathbf{False}\big)$

● **Skip**

end

section $\mathit{FlatBufferMissionMethChan}$ **parents** $\mathit{scj\_prelude}, \mathit{GlobalTypes}, \mathit{MissionId}, \mathit{SchedulableId}$

channel $\mathit{bufferEmptyCall} : \mathit{MissionID}$
channel $\mathit{bufferEmptyRet} : \mathit{MissionID} \times \mathbb{B}$

channel $\mathit{cleanUpCall} : \mathit{MissionID}$
channel $\mathit{cleanUpRet} : \mathit{MissionID} \times \mathbb{B}$

channel $\mathit{writeCall} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadID} \times \mathbb{Z}$
channel $\mathit{writeRet} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadID}$

channel $\mathit{readCall} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadID}$
channel $\mathit{readRet} : \mathit{MissionID} \times \mathit{SchedulableID} \times \mathit{ThreadID} \times \mathbb{Z}$

5.2 Schedulables of FlatBufferMission

section *ReaderApp* **parents** *ManagedThreadChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels* , *MissionMethChan*, *FlatBufferMissionMethChan*, *ObjectIds*, *ThreadIds*

$$\begin{array}{l}\textbf{process}\ ReaderApp \mathrel{\widehat=} \\ \quad fbMission : MissionID \bullet \textbf{begin}\end{array}$$

$$\begin{array}{l}
Run \mathrel{\widehat=} \\
\left(\begin{array}{l}
runCall \,.\, ReaderSID \longrightarrow \\
\left(\left(\left(\begin{array}{l}
\mu X \bullet \\
\left(\begin{array}{l}
terminationPendingCall \,.\, fbMission \longrightarrow \\
terminationPendingRet \,.\, fbMission \,?\, terminationPending \longrightarrow \\
\textbf{var}\ loopVar : \mathbb{B} \bullet loopVar := (\lnot\ terminationPending); \\
\textbf{if}\ (loopVar = \textbf{True}) \longrightarrow \\
\qquad \left(\begin{array}{l}
\textbf{var}\ result : \mathbb{Z} \bullet result := -1; \\
\left(\begin{array}{l}
binder\_readCall \,.\, fbMission \,.\, ReaderSID \,.\, ReaderTID \longrightarrow \\
binder\_readRet \,.\, fbMission \,.\, ReaderSID \,.\, ReaderTID \,?\, read \longrightarrow
\end{array}\right);
\end{array}\right); \ X \\
\ [\!]\ (loopVar = \textbf{False}) \longrightarrow \textbf{Skip} \\
\textbf{fi}
\end{array}\right)
\end{array}\right)\right)\right); \\
runRet \,.\, ReaderSID \longrightarrow \\
\textbf{Skip}
\end{array}\right)
\end{array}$$

$$\begin{array}{l}
Methods \mathrel{\widehat=} \\
\big(Run\big)\ ; \ Methods
\end{array}$$

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

end

section *WriterApp* **parents** *ManagedThreadChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*
, *MissionMethChan*, *FlatBufferMissionMethChan*, *ObjectIds*, *ThreadIds*

$$\mathbf{process}\ \mathit{WriterApp} \mathrel{\widehat=} \\ \quad \mathit{fbMission} : \mathit{MissionID} \bullet \mathbf{begin}$$

$$
\mathit{Run} \mathrel{\widehat=}
\left(\begin{array}{l}
\mathit{runCall} \,.\, \mathit{WriterSID} \longrightarrow \\
\left(\begin{array}{l}
\mathbf{var}\ i : \mathbb{Z} \bullet i := 1; \\
\left(\begin{array}{l}
\mu X \bullet \\
\left(\begin{array}{l}
\mathit{terminationPendingCall} \,.\, \mathit{fbMission} \longrightarrow \\
\mathit{terminationPendingRet} \,.\, \mathit{fbMission} \,?\, \mathit{terminationPending} \longrightarrow \\
\mathbf{var}\ \mathit{loopVar} : \mathbb{B} \bullet \mathit{loopVar} := (\lnot\ \mathit{terminationPending}); \\
\mathbf{if}\ (\mathit{loopVar} = \mathbf{True}) \longrightarrow \\
\qquad \left(\begin{array}{l}
\left(\begin{array}{l}
\mathit{binder\_writeCall} \,.\, \mathit{fbMission} \,.\, \mathit{WriterSID} \,.\, \mathit{WriterTID} \,!\, i \longrightarrow \\
\mathit{binder\_writeRet} \,.\, \mathit{fbMission} \,.\, \mathit{WriterSID} \,.\, \mathit{WriterTID} \longrightarrow \\
\mathbf{Skip}
\end{array}\right) ; \\
i := i + 1; \\
\mathbf{if}\ (i \geq 5) \longrightarrow \\
\qquad \left(\begin{array}{l}
\mathit{requestTerminationCall} \,.\, \mathit{fbMission} \longrightarrow \\
\mathit{requestTerminationRet} \,.\, \mathit{fbMission} \,?\, \mathit{requestTermination} \longrightarrow
\end{array}\right) \\
\square\ \lnot\ (i \geq 5) \longrightarrow \mathbf{Skip} \\
\mathbf{fi}\ ;
\end{array}\right) ;\ X \\
\square\ (\mathit{loopVar} = \mathbf{False}) \longrightarrow \mathbf{Skip} \\
\mathbf{fi}
\end{array}\right)
\end{array}\right)
\end{array}\right) ; \\
\mathit{runRet} \,.\, \mathit{WriterSID} \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
$$

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

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

end