twoSequentialMissions Report

4th October 2016

1 ID Files

1.1 MissionIds

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

$$
\begin{array}{|l}
MissionAMID : MissionID \\
MissionBMID : MissionID \\
\hline
distinct \langle nullMissionId, MissionAMID, MissionBMID \rangle
\end{array}
$$

1.2 SchedulablesIds

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

$$\begin{array}{|l}
mainSequencerSID : SchedulableID \\
MT1SID : SchedulableID \\
MT2SID : SchedulableID \\
\hline
distinct \langle nullSequencerId, nullSchedulableId, mainSequencerSID, \\
MT1SID, MT2SID \rangle
\end{array}$$

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 $TierSync ==$
$\{\!| start\_mission . MissionA, done\_mission . MissionA,$
$done\_safeletFW, done\_toplevel\_sequencer |\!\}$

channelset $TierSync ==$
$\{\!| start\_mission . MissionB, done\_mission . MissionB,$
$done\_safeletFW, done\_toplevel\_sequencer |\!\}$

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 $SafeltAppSync \mathrel{\widehat{=}}$
$\{\!| getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end\_safelet\_app |\!\}$

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 $LockingSync ==$
$\{\!| lockAcquired, startSyncMeth, endSyncMeth, waitCall, waitRet, notify, isInterruptedCall, isInterruptedRet,$
$interruptedCall, interruptedRet, done\_toplevel\_sequencer, get\_priorityLevel |\!\}$

2.2 Locking

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

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

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

process *Locking* $\widehat{=}$ *Threads* $[\![$ *ThreadSync* $]\!]$ *Objects*

2.3 Program

$$\begin{array}{l}
\textbf{section } \mathit{Program} \textbf{ parents } \mathit{scj\_prelude}, \mathit{MissionId}, \mathit{MissionIds}, \\
\quad \mathit{SchedulableId}, \mathit{SchedulableIds}, \mathit{MissionChan}, \mathit{SchedulableMethChan}, \mathit{MissionFW}, \\
\quad \mathit{SafeletFW}, \mathit{TopLevelMissionSequencerFW}, \mathit{NetworkChannels}, \mathit{ManagedThreadFW}, \\
\quad \mathit{SchedulableMissionSequencerFW}, \mathit{PeriodicEventHandlerFW}, \mathit{OneShotEventHandlerFW}, \\
\quad \mathit{AperiodicEventHandlerFW}, \mathit{ObjectFW}, \mathit{ThreadFW}, \\
\quad \mathit{MyAppApp}, \mathit{mainSequencerApp}, \mathit{MissionAApp}, \mathit{MT1App}, \mathit{MissionBApp}, \mathit{MT2App}
\end{array}$$

$$\begin{array}{l}
\textbf{process } \mathit{ControlTier} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{SafeletFW} \\
\quad \llbracket \mathit{ControlTierSync} \rrbracket \\
\mathit{TopLevelMissionSequencerFW}(\mathit{mainSequencer})
\end{array}\right)
\end{array}$$

$$\begin{array}{l}
\textbf{process } \mathit{Tier0} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{MissionFW}(\mathit{MissionAID}) \\
\quad \llbracket \mathit{MissionSync} \rrbracket \\
(\mathit{ManagedThreadFW}(\mathit{MT1ID}))
\end{array}\right) \\
\quad \llbracket \mathit{ClusterSync} \rrbracket \\
\left(\begin{array}{l}
\mathit{MissionFW}(\mathit{MissionBID}) \\
\quad \llbracket \mathit{MissionSync} \rrbracket \\
(\mathit{ManagedThreadFW}(\mathit{MT2ID}))
\end{array}\right)
\end{array}$$

$$\begin{array}{l}
\textbf{process } \mathit{Framework} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{ControlTier} \\
\quad \llbracket \mathit{TierSync} \rrbracket \\
(\mathit{Tier0})
\end{array}\right)
\end{array}$$

$$\begin{array}{l}
\textbf{process } \mathit{Application} \mathrel{\widehat=} \\
\left(\begin{array}{l}
\mathit{MyAppApp} \\
\interleave \\
\mathit{mainSequencerApp} \\
\interleave \\
\mathit{MissionAApp} \\
\interleave \\
\mathit{MT1App} \\
\interleave \\
\mathit{MissionBApp} \\
\interleave \\
\mathit{MT2App}
\end{array}\right)
\end{array}$$

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

3 Safelet

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

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

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

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

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

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

$\mathbf{end}$

4 Top Level Mission Sequencer

$$\begin{array}{l}
\mathbf{section}\ mainSequencerApp\ \mathbf{parents}\ TopLevelMissionSequencerChan, \\
\quad MissionId, MissionIds, SchedulableId, SchedulableIds, mainSequencerClass, MethodCallBindingChannels
\end{array}$$

$$\begin{array}{l}
\mathbf{process}\ mainSequencerApp \mathrel{\widehat=} \\
\quad name : String \bullet \mathbf{begin}
\end{array}$$

$$\begin{array}{|l}
\hline
State \\
\quad this : \mathbf{ref}\ mainSequencerClass \\
\hline
\end{array}$$

$$\mathbf{state}\ State$$

$$\begin{array}{|l}
\hline
Init \\
\quad State\,' \\
\hline
\quad this' = \mathbf{new}\ mainSequencerClass() \\
\hline
\end{array}$$

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

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

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

end

$\mathbf{section}\ mainSequencerClass\ \mathbf{parents}\ scj\_prelude, SchedulableId, SchedulableIds, SafeletChan$
$, MethodCallBindingChannels, MissionId, MissionIds$

$\mathbf{class}\ mainSequencerClass \mathrel{\widehat=} \mathbf{begin}$

$$\begin{array}{|l}
\mathbf{state}\ State \\
\hline
releases : \mathbb{Z} \\
\hline
\end{array}$$

$\mathbf{state}\ State$

$$\begin{array}{|l}
\mathbf{initial}\ Init \\
\hline
State\,' \\
\hline
releases = 0 \\
\hline
\end{array}$$

$$\begin{array}{l}
\mathbf{protected}\ getNextMission \mathrel{\widehat=} \mathbf{var}\ ret : MissionID \bullet \\
\left(\begin{array}{l}
\mathbf{if}\ releases = 0 \longrightarrow \\
\qquad \left(\begin{array}{l} releases := releases + 1; \\ ret := MissionAMID \end{array}\right) \\
[\!]\ \neg\ releases = 0 \longrightarrow \\
\qquad \mathbf{if}\ releases = 1 \longrightarrow \\
\qquad\qquad \left(\begin{array}{l} releases := releases + 1; \\ ret := MissionBMID \end{array}\right) \\
\qquad [\!]\ \neg\ releases = 1 \longrightarrow \\
\qquad\qquad \left(ret := nullMissionId\right) \\
\qquad \mathbf{fi} \\
\mathbf{fi}
\end{array}\right)
\end{array}$$

$\bullet\ \mathbf{Skip}$

$\mathbf{end}$

5 Missions

5.1 MissionA

section *MissionAApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MissionAMethChan*
, *MethodCallBindingChannels*

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

$$
\begin{array}{l}
InitializePhase \mathrel{\widehat{=}} \\
\left(\begin{array}{l}
initializeCall\,.\,MissionAMID \longrightarrow \\
register\,!\,MT1SID\,!\,MissionAMID \longrightarrow \\
initializeRet\,.\,MissionAMID \longrightarrow \\
\mathbf{Skip}
\end{array}\right)
\end{array}
$$

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

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

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

end

5.2 Schedulables of MissionA

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

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

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

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

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

end

5.3 MissionB

section *MissionBApp* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MissionBMethChan*
, *MethodCallBindingChannels*

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

$$InitializePhase \mathrel{\widehat{=}} \left(\begin{array}{l} initializeCall \,.\, MissionBMID \longrightarrow \\ register \,!\, MT2SID \,!\, MissionBMID \longrightarrow \\ initializeRet \,.\, MissionBMID \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

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

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

end

5.4 Schedulables of MissionB

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

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

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

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

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

end