mission2 Report

4th October 2016

1 ID Files

1.1 MissionIds

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

$$MissionAMID : MissionID$$

$$distinct\langle nullMissionId, MissionAMID\rangle$$

1.2 SchedulablesIds

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

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

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 *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* |}

2.2 MethodCallBinder

section *MethodCallBindingChannels* **parents** *scj_prelude*, *GlobalTypes*, *FrameworkChan*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *ThreadIds*

channel $binder\_systemActionCall : MissionID \times SchedulableID$
channel $binder\_systemActionRet : MissionID \times SchedulableID$

$systemActionLocs == \{MissionAMID\}$
$systemActionCallers == \{MTSID\}$

channelset $MethodCallBinderSync == \{\!|\ done\_toplevel\_sequencer, binder\_systemActionCall, binder\_systemActionRet\ |\!\}$

section *MethodCallBinder* **parents** *scj_prelude*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*
, *MissionAMethChan*

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

$$systemAction\_MethodBinder \mathrel{\widehat=} \left(\begin{array}{l} binder\_systemActionCall\,?\,loc : (loc \in systemActionLocs)\,?\,caller : (caller \in systemActionCallers) \longrightarrow \\ systemActionCall\,.\,loc\,.\,caller \longrightarrow \\ systemActionRet\,.\,loc\,.\,caller \longrightarrow \\ binder\_systemActionRet\,.\,loc\,.\,caller \longrightarrow \\ systemAction\_MethodBinder \end{array}\right)$$

$$BinderActions \mathrel{\widehat=} \left(systemAction\_MethodBinder\right)$$

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

end

2.3 Locking

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

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

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

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*,
MyAppApp, *mainSequencerApp*, *MissionAApp*, *MTApp*, *OSEHApp*

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

process $Tier0 \mathrel{\widehat=}$
$$\left(\begin{array}{l} MissionFW(MissionAID) \\ \quad [\![MissionSync]\!] \\ \left(\begin{array}{l} ManagedThreadFW(MTID) \\ \quad [\![SchedulablesSync]\!] \\ OneShotEventHandlerFW(OSEHID, (time(60,0)), (time(5,0), null)) \end{array}\right) \end{array}\right)$$

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

process $Application \mathrel{\widehat=}$
$$\left(\begin{array}{l} MyAppApp \\ ||| \\ mainSequencerApp \\ ||| \\ MissionAApp \\ ||| \\ MTApp(MissionAID) \\ ||| \\ OSEHApp(MissionAID) \end{array}\right)$$

process $Bound\_Application \mathrel{\widehat=} Application \ [\![\ MethodCallBinderSync \ ]\!] \ MethodCallBinder$
process $Program \mathrel{\widehat=} (\ Framework \ [\![\ AppSync \ ]\!] \ Bound\_Application\ ) \ [\![\ LockingSync \ ]\!] \ Locking$

3 Safelet

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

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

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

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

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

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

end

4 Top Level Mission Sequencer

section *mainSequencerApp* **parents** *TopLevelMissionSequencerChan*, *MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *mainSequencerClass*, *MethodCallBindingChannels*

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

State

$this$: **ref** *mainSequencerClass*

state *State*

Init

$State'$

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

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

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

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

end

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

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

state *State*

$$notReleased : \mathbb{B}$$

state *State*

initial *Init*

$$State'$$

$$notReleased = \mathbf{True}$$

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

$$\left(\begin{array}{l}
\mathbf{if}\ notReleased = \mathbf{True} \longrightarrow \\
\quad \left(\begin{array}{l}
\mathbf{var}\ mission : MissionID \bullet mission := MissionAMID; \\
this\,.\,notReleased := \mathbf{False}; \\
ret := mission
\end{array}\right) \\
[]\ \neg\ notReleased = \mathbf{True} \longrightarrow \\
\quad \left(ret := nullMissionId\right) \\
\mathbf{fi}
\end{array}\right)$$

• **Skip**

end

5 Missions

5.1 MissionA

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

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

State

$this : \textbf{ref}\ MissionAClass$

state *State*

Init

$State'$

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

$$InitializePhase \mathrel{\widehat=} \begin{pmatrix} initializeCall\ .\ MissionAMID \longrightarrow \\ register\ !\ OSEHSID\ !\ MissionAMID \longrightarrow \\ register\ !\ MTSID\ !\ MissionAMID \longrightarrow \\ initializeRet\ .\ MissionAMID \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

$$systemActionMeth \mathrel{\widehat=} \begin{pmatrix} systemActionCall\ .\ MissionAMID\ ?\ caller \longrightarrow \\ this\ .\ systemAction(); \\ systemActionRet\ .\ MissionAMID\ .\ caller \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

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

end

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

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

public *systemAction* $\widehat{=}$
Skip

• **Skip**

end

5.2 Schedulables of MissionA

section *MTApp* **parents** *ManagedThreadChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*, *MissionAMethChan*

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

$$Run \mathrel{\widehat=}
\left(\begin{array}{l}
runCall\,.\,MTSID \longrightarrow \\
\left(\begin{array}{l}
binder_systemActionCall\,.\,controllingMission\,.\,MTSID \longrightarrow \\
binder_systemActionRet\,.\,controllingMission\,.\,MTSID \longrightarrow \\
\textbf{Skip}
\end{array}\right) ; \\
runRet\,.\,MTSID \longrightarrow \\
\textbf{Skip}
\end{array}\right)$$

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

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

end

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

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

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

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

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

end