aircraft

Tight Rope v0.6

October 20, 2015

1 Network

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* . , *done_mission* . ,
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* ==
$\bigcup\{$*SafeltAppSync*, *MissionSequencerAppSync*, *MissionAppSync*,
MTAppSync, *OSEHSync*, *APEHSync*,
$\{\!|\,$ *getSequencer*, *end_mission_app*, *end_managedThread_app*,
setCeilingPriority, *requestTerminationCall*, *requestTerminationRet*, *terminationPendingCall*,
terminationPendingRet, *handleAsyncEventCall*, *handleAsyncEventRet* $\,|\!\}\}$

channelset *ObjectSync* ==
$\{\!|\;|\!\}$

channelset *ThreadSync* ==
$\{\!|\;|\!\}$

channelset *LockingSync* ==
$\{\!|\,$ *lockAcquired*, *startSyncMeth*, *endSyncMeth*, *waitCall*, *waitRet*, *notify* $\,|\!\}$

section *Program* **parents** *scj_prelude*, *MissionId*, *MissionIds*,
    SchedulableId, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MissionFW*,
    SafeletFW, *TopLevelMissionSequencerFW*, *NetworkChannels*, *ManagedThreadFW*,
    SchedulableMissionSequencerFW, *PeriodicEventHandlerFW*, *OneShotEventHandlerFW*,
    AperiodicEventHandlerFW, *ACSafeletApp*, *MainMissionSequencerApp*,
    ObjectFW, *ThreadFW*,   *MainMissionApp*,

process *ControlTier* $\widehat{=}$
$$\begin{pmatrix} SafeletFW \\ \quad [\![ControlTierSync]\!] \\ TopLevelMissionSequencerFW\,(MainMissionSequencer) \end{pmatrix}$$

process *Tier0* $\widehat{=}$
$$\begin{pmatrix} MissionFW\,(MainMission) \\ \quad [\![MissionSync]\!] \\ (\end{pmatrix}$$

process *Framework* $\widehat{=}$
$$\begin{pmatrix} ControlTier \\ \quad [\![TierSync]\!] \\ (\,Tier0\,) \end{pmatrix}$$

process *Application* $\widehat{=}$
$$\begin{pmatrix} ACSafeletApp \\ ||| \\ MainMissionSequencerApp \\ ||| \\ MainMissionApp \\ ||| \end{pmatrix}$$

Locking $\widehat{=}$
$$\begin{pmatrix} (\\ ||| \\ \begin{pmatrix} ObjectFW\,(ACSafeletObject) \\ \quad [\![ObjectSync]\!] \\ ObjectFW\,(MainMissionObject) \end{pmatrix} \end{pmatrix}$$

process *Program* $\widehat{=}$ *Framework* $[\![\ AppSync\ ]\!]$ *Application* $[\![\ LockingSync\ ]\!]$ *Locking*

2 ID Files

2.1 MissionIds

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

$MainMission : MissionID$

$distinct\langle nullMissionId, MainMission\rangle$

2.2 SchedulablesIds

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

$MainMissionSequencer : SchedulableID$

$distinct\langle nullSequencerId, nullSchedulableId, \rangle$

2.3 ThreadIds

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

$distinct\langle SafeletThreadId, nullThreadId, \rangle$

2.4 ObjectIds

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

$ACSafeletObject : ObjectID$
$MainMissionObject : ObjectID$

$distinct\langle ACSafeletObject, MainMissionObject\rangle$

3 Safelet

section *ACSafeletApp* **parents** *scj_prelude, SchedulableId, SchedulableIds, SafeletChan*

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

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

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

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

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

end

4 Top Level Mission Sequencer

section *MainMissionSequencerApp* **parents** *TopLevelMissionSequencerChan,*
    MissionId, MissionIds, SchedulableId, MainMissionSequencerClass

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

State
this : **ref** *MainMissionSequencerClass*

---

state *State*

Init
State′

---
this′ = **new** *MainMissionSequencerClass*()

GetNextMission $\widehat{=}$ **var** *ret* : *MissionID* •
$$\begin{pmatrix} getNextMissionCall . MainMissionSequencer \longrightarrow \\ ret := this . getNextMission(); \\ getNextMissionRet . MainMissionSequencer ! ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$$

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

• (*Init* ; *Methods*) $\triangle$ (*end_sequencer_app . MainMissionSequencer* $\longrightarrow$ **Skip**)

end

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

---
state *State* ─────────────────────────────
  returnedMission : $\mathbb{B}$

---

state *State*

---
initial *Init* ─────────────────────────────
  State$'$

  ─────────
  returnedMission$'$ = *false*

---

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

$$
\begin{pmatrix}
\textbf{if} \, (\neg \, returnedMission = \textbf{True}) \longrightarrow \\
\quad \begin{pmatrix} this \, . \, returnedMission := true; \\ ret := MainMission \end{pmatrix} \\
[\!] \, \neg \, (\neg \, returnedMission = \textbf{True}) \longrightarrow \\
\quad \begin{pmatrix} ret := nullMissionId \end{pmatrix} \\
\textbf{fi}
\end{pmatrix}
$$

$\bullet$ **Skip**

end

5 Missions

5.1 MainMission

section *MainMissionApp* **parents** *scj_prelude, MissionId, MissionIds,*
  SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MainMissionClass   , MainMissionMethChan

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

State ─────────────────────────────────
  this : **ref** *MainMissionClass*
───────────────────────────────────────

state *State*

Init ─────────────────────────────────
  State′
  ─────────
  this′ = **new** *MainMissionClass*()
───────────────────────────────────────

InitializePhase $\widehat{=}$
$\begin{pmatrix} initializeCall \, . \, MainMission \longrightarrow \\ initializeRet \, . \, MainMission \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

CleanupPhase $\widehat{=}$
$\begin{pmatrix} cleanupMissionCall \, . \, MainMission \longrightarrow \\ cleanupMissionRet \, . \, MainMission \, ! \, \textbf{False} \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

getAirSpeedMeth $\widehat{=}$ **var** *ret* : *double* •
$\begin{pmatrix} getAirSpeedCall \, . \, MainMission \longrightarrow \\ ret := this \, . \, getAirSpeed(); \\ getAirSpeedRet \, . \, MainMission \, ! \, ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

getAltitudeMeth $\widehat{=}$ **var** *ret* : *double* •
$\begin{pmatrix} getAltitudeCall \, . \, MainMission \longrightarrow \\ ret := this \, . \, getAltitude(); \\ getAltitudeRet \, . \, MainMission \, ! \, ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

getCabinPressureMeth $\widehat{=}$ **var** *ret* : *double* •
$\begin{pmatrix} getCabinPressureCall \, . \, MainMission \longrightarrow \\ ret := this \, . \, getCabinPressure(); \\ getCabinPressureRet \, . \, MainMission \, ! \, ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

getEmergencyOxygenMeth $\widehat{=}$ **var** *ret* : *double* •
$\begin{pmatrix} getEmergencyOxygenCall \, . \, MainMission \longrightarrow \\ ret := this \, . \, getEmergencyOxygen(); \\ getEmergencyOxygenRet \, . \, MainMission \, ! \, ret \longrightarrow \\ \textbf{Skip} \end{pmatrix}$

$getFuelRemainingMeth \mathrel{\widehat{=}} \mathbf{var}\ ret : double \bullet$
$$\begin{pmatrix} getFuelRemainingCall\,.\,MainMission \longrightarrow \\ ret := this\,.\,getFuelRemaining(); \\ getFuelRemainingRet\,.\,MainMission\,!\,ret \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$getHeadingMeth \mathrel{\widehat{=}} \mathbf{var}\ ret : double \bullet$
$$\begin{pmatrix} getHeadingCall\,.\,MainMission \longrightarrow \\ ret := this\,.\,getHeading(); \\ getHeadingRet\,.\,MainMission\,!\,ret \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setAirSpeedMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setAirSpeedCall\,.\,MainMission\,?\,airSpeed \longrightarrow \\ this\,.\,setAirSpeed(airSpeed); \\ setAirSpeedRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setAltitudeMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setAltitudeCall\,.\,MainMission\,?\,altitude \longrightarrow \\ this\,.\,setAltitude(altitude); \\ setAltitudeRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setCabinPressureMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setCabinPressureCall\,.\,MainMission\,?\,cabinPressure \longrightarrow \\ this\,.\,setCabinPressure(cabinPressure); \\ setCabinPressureRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setEmergencyOxygenMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setEmergencyOxygenCall\,.\,MainMission\,?\,emergencyOxygen \longrightarrow \\ this\,.\,setEmergencyOxygen(emergencyOxygen); \\ setEmergencyOxygenRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setFuelRemainingMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setFuelRemainingCall\,.\,MainMission\,?\,fuelRemaining \longrightarrow \\ this\,.\,setFuelRemaining(fuelRemaining); \\ setFuelRemainingRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$setHeadingMeth \mathrel{\widehat{=}}$
$$\begin{pmatrix} setHeadingCall\,.\,MainMission\,?\,heading \longrightarrow \\ this\,.\,setHeading(heading); \\ setHeadingRet\,.\,MainMission \longrightarrow \\ \mathbf{Skip} \end{pmatrix}$$

$$
Methods \mathrel{\widehat{=}} \left(\begin{array}{l} InitializePhase \\ \Box \\ CleanupPhase \\ \Box \\ getAirSpeedMeth \\ \Box \\ getAltitudeMeth \\ \Box \\ getCabinPressureMeth \\ \Box \\ getEmergencyOxygenMeth \\ \Box \\ getFuelRemainingMeth \\ \Box \\ getHeadingMeth \\ \Box \\ setAirSpeedMeth \\ \Box \\ setAltitudeMeth \\ \Box \\ setCabinPressureMeth \\ \Box \\ setEmergencyOxygenMeth \\ \Box \\ setFuelRemainingMeth \\ \Box \\ setHeadingMeth \end{array}\right) \; ; \; Methods
$$

● $(Init \; ; \; Methods) \mathbin{\triangle} (end\_mission\_app . MainMission \longrightarrow \mathbf{Skip})$

end

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

---
state *State*
> *ALTITUDE_READING_ON_GROUND* : *double*
> *cabinPressure* : *double*
> *emergencyOxygen* : *double*
> *fuelRemaining* : *double*
> *altitude* : *double*
> *airSpeed* : *double*
> *heading* : *double*
---

state *State*

---
initial *Init*
> *State$'$*
>
> *ALTITUDE_READING_ON_GROUND$'$* = 0.0
---

public *getAirSpeed* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *airSpeed* $)$

public *getAltitude* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *altitude* $)$

public *getCabinPressure* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *cabinPressure* $)$

public *getEmergencyOxygen* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *emergencyOxygen* $)$

public *getFuelRemaining* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *fuelRemaining* $)$

public *getHeading* $\widehat{=}$ **var** *ret* : *double* $\bullet$
$($*ret* := *heading* $)$

public *setAirSpeed* $\widehat{=}$
$($*this* . *this.airSpeed* := *airSpeed* $)$

public *setAltitude* $\widehat{=}$
$($*this* . *this.altitude* := *altitude* $)$

public *setCabinPressure* $\widehat{=}$
$($*this* . *this.cabinPressure* := *cabinPressure* $)$

public *setEmergencyOxygen* $\widehat{=}$
$($*this* . *this.emergencyOxygen* := *emergencyOxygen* $)$

public *setFuelRemaining* $\widehat{=}$
$\big(this\,.\,this.fuelRemaining := fuelRemaining\big)$

public *setHeading* $\widehat{=}$
$\big(this\,.\,this.heading := heading\big)$

- **Skip**

end

section *MainMissionMethChan* **parents** *scj_prelude*, *GlobalTypes*, *MissionId*, *SchedulableId*

channel *getAirSpeedCall* : *MissionID*
channel *getAirSpeedRet* : *MissionID* × *double*

channel *getAltitudeCall* : *MissionID*
channel *getAltitudeRet* : *MissionID* × *double*

channel *getCabinPressureCall* : *MissionID*
channel *getCabinPressureRet* : *MissionID* × *double*

channel *getEmergencyOxygenCall* : *MissionID*
channel *getEmergencyOxygenRet* : *MissionID* × *double*

channel *getFuelRemainingCall* : *MissionID*
channel *getFuelRemainingRet* : *MissionID* × *double*

channel *getHeadingCall* : *MissionID*
channel *getHeadingRet* : *MissionID* × *double*

channel *setAirSpeedCall* : *MissionID* × *double*
channel *setAirSpeedRet* : *MissionID*

channel *setAltitudeCall* : *MissionID* × *double*
channel *setAltitudeRet* : *MissionID*

channel *setCabinPressureCall* : *MissionID* × *double*
channel *setCabinPressureRet* : *MissionID*

channel *setEmergencyOxygenCall* : *MissionID* × *double*
channel *setEmergencyOxygenRet* : *MissionID*

channel *setFuelRemainingCall* : *MissionID* × *double*
channel *setFuelRemainingRet* : *MissionID*

channel *setHeadingCall* : *MissionID* × *double*
channel *setHeadingRet* : *MissionID*

5.2 Schedulables of