$\mathbf{channelset}\ SafeltAppSync ==$
$\lbrace\!| getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end\_safelet\_app |\!\rbrace$

$\mathbf{channelset}\ MyAppChanSet ==$
$\lbrace\!| getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end\_safelet\_app |\!\rbrace$

$\mathbf{process}\ MyApp \mathrel{\widehat=}$

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

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

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

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

$\mathbf{channelset}\ MissionSequencerAppSync ==$
$\{\!|\ getNextMissionCall, getNextMissionRet, end\_sequencer\_app\ |\!\}$

$\mathbf{channelset}\ MySequencerAppChanSet == MissionSequencerAppSync$

$\mathbf{process}\ MainSequencerApp \mathrel{\widehat=}$

$GetNextMission \mathrel{\widehat=}$
$$\left(\begin{array}{l} getNextMissionCall\,.\,[TopLevelSequencer] \longrightarrow \\ getNextMissionRet\,.\,[TopLevelSequencer]\,!\,[MainMission] \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

$\bullet\ (Methods) \setminus MissionSequencerAppStateSync \vartriangle (end\_sequencer\_app\,.\,[TopLevelSequencer] \longrightarrow \mathbf{Skip})$

$\mathbf{channelset}\ \mathit{MissionAppSync} ==$
$\{\!|\ \mathit{initializeCall}, \mathit{register}, \mathit{initializeRet}, \mathit{cleanupMissionCall}, \mathit{cleanupMissionRet}\ |\!\}$

$\mathbf{channelset}\ \mathit{MissionAppChanSet} ==$
$\{\!|\ \mathit{initializeCall}, \mathit{initializeRet}, \mathit{cleanupMissionCall}, \mathit{cleanupMissionRet}, \mathit{register} . \mathit{NestedMissionSequencerId} . [\mathit{Mission}]\ |\!\}$

$\mathbf{process}\ \mathit{MainMissionApp} \mathrel{\widehat=}$

$$\mathit{InitializePhase} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{initializeCall} . [\mathit{Mission}] \longrightarrow \\ \mathit{register}\ !\ [\mathit{SchedulableObject}]\ !\ [\mathit{Mission}] \longrightarrow \\ \mathit{initializeRet} . [\mathit{Mission}] \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$$\mathit{CleanupPhase} \mathrel{\widehat=} \left(\begin{array}{l} \mathit{cleanupMissionCall} . [\mathit{Mission}] \longrightarrow \\ \mathit{cleanupMissionRet} . [\mathit{Mission}]\ ?\ \mathit{false} \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

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

$\bullet\ (\mathit{Methods}) \bigtriangleup (\mathit{end\_mission\_app} . [\mathit{Mission}] \longrightarrow \mathbf{Skip})$

process $HandlerApp \mathrel{\widehat=}$

$Methods \mathrel{\widehat=}$
$handlerAsyncEvent \ ; \ Methods$

$handlerAsyncEvent \mathrel{\widehat=}$
$$\left(\begin{array}{l} handleAsyncEventCall \,.\, [Handler] \longrightarrow \\ handleAsyncEventRet \,.\, [Handler] \longrightarrow \\ \mathbf{Skip} \end{array}\right)$$

$\bullet \ (Methods) \mathbin{\triangle} (end\_[handlerType]EventHandler\_app \,.\, [Handler] \longrightarrow \mathbf{Skip})$

process $\mathit{ManagedThreadApp} \mathrel{\widehat=}$

$\mathit{Methods} \mathrel{\widehat=}$
$\mathit{Run} \;;\; \mathit{Methods}$

$\mathit{Run} \mathrel{\widehat=}$
$$\left(\begin{array}{l}\mathit{runCall} \,.\, [\mathit{ManagedThread}] \longrightarrow \\ \mathit{runRet} \,.\, [\mathit{ManagedThread}] \longrightarrow \\ \mathbf{Skip}\end{array}\right)$$

$\bullet\; (\mathit{Methods}) \bigtriangleup (\mathit{end\_managedThread\_app} \,.\, [\mathit{ManagedThread}] \longrightarrow \mathbf{Skip})$