MODULE *P0ResStore*

Resilient store hosted at *PROG_HOME* and is assumed be always available See *FinishResilientPlace*0.*x*10 for the actual implementation

EXTENDS *Integers*, *Sequences*

CONSTANTS *PLACE*, *MXFINISHES*, *PROG_HOME*, *MXTHREADS*, *NBLOCKS*, *MXSTMTS*

VARIABLES *fstates*, *msgs*, *pstate*, *program*, *aseq*, *fseq*, *mseq*,
readyQ, *thrds*, *ppProgram*, *ppcurStmt*, *incPar*, *decPar*,
killed, *killedCnt*, *p0fstates*, *pendingAct*, *isDead*, *p0dead*, *p0adoptSet*, *p0state*, *p0convSet*

INSTANCE *Commons*

$$
\begin{aligned}
&\mathit{AddException}(\mathit{fid}, \mathit{pfid}, \mathit{root}, \mathit{rootPlace}, \mathit{src}, \mathit{target}) \triangleq \\
&\quad \land\ \mathit{p0fstates}' = [\mathit{p0fstates}\ \text{EXCEPT}\ ![\mathit{fid}].\mathit{id} = \mathit{fid}, \\
&\qquad\qquad ![\mathit{fid}].\mathit{parent} = \mathit{pfid}, \\
&\qquad\qquad ![\mathit{fid}].\mathit{gfsRoot} = \mathit{root}, \\
&\qquad\qquad ![\mathit{fid}].\mathit{gfsRootPlace} = \mathit{rootPlace}, \\
&\qquad\qquad ![\mathit{fid}].\mathit{excs} = \mathit{Append}(@, [\mathit{err} \mapsto \text{“DPE”}, \mathit{from} \mapsto \mathit{target}])]
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{AddTransitAdopted}(\mathit{fid}, \mathit{pfid}, \mathit{root}, \mathit{rootPlace}, \mathit{src}, \mathit{target}) \triangleq \\
&\quad \text{LET}\ \mathit{adopter} \triangleq \mathit{p0fstates}[\mathit{fid}].\mathit{adopterId} \\
&\quad \text{IN}\ \ \mathit{p0fstates}' = [\mathit{p0fstates}\ \text{EXCEPT}\ ![\mathit{adopter}].\mathit{numActive} = @ + 1, \\
&\qquad\qquad\qquad\qquad ![\mathit{adopter}].\mathit{transitAdopted}[\mathit{src}][\mathit{target}] = @ + 1]
\end{aligned}
$$

initialize parent too if not initialized

$$
\begin{aligned}
&\mathit{AddTransit}(\mathit{fid}, \mathit{pfid}, \mathit{root}, \mathit{rootPlace}, \mathit{src}, \mathit{target}) \triangleq \\
&\quad \text{IF}\ \mathit{p0fstates}[\mathit{fid}].\mathit{adopterId} = \mathit{NotID} \\
&\qquad \text{THEN IF}\ \mathit{p0fstates}[\mathit{fid}].\mathit{id} = \mathit{NotID}\ \text{not yet initialized} \\
&\qquad\qquad \text{THEN}\ \ \text{IF}\ \mathit{pfid} > \mathit{NoParent} \land \mathit{p0fstates}[\mathit{pfid}].\mathit{id} = \mathit{NotID} \\
&\qquad\qquad\qquad \text{THEN}\ \mathit{p0fstates}' = [\mathit{p0fstates}\ \text{EXCEPT}\ ![\mathit{fid}].\mathit{id} = \mathit{fid}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{parent} = \mathit{pfid}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{gfsRoot} = \mathit{root}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{gfsRootPlace} = \mathit{rootPlace}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{numActive} = @ + 2, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{transit}[\mathit{src}][\mathit{target}] = @ + 1, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{live}[\mathit{rootPlace}] = @ + 1, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{id} = \ \mathit{pfid}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{parent} = \mathit{fstates}[\mathit{pfid}].\mathit{parent}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{gfsRoot} = \mathit{fstates}[\mathit{pfid}].\mathit{root}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{gfsRootPlace} = \mathit{fstates}[\mathit{fstates}[\mathit{pfid}].\mathit{root}].\mathit{here}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{numActive} = 1, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{pfid}].\mathit{live}[\mathit{fstates}[\mathit{pfid}].\mathit{here}] = 1]\ \text{root finish's task} \\
&\qquad\qquad\qquad \text{ELSE}\ \ \mathit{p0fstates}' = [\mathit{p0fstates}\ \text{EXCEPT}\ ![\mathit{fid}].\mathit{id} = \mathit{fid}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{parent} = \mathit{pfid}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{gfsRoot} = \mathit{root}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{gfsRootPlace} = \mathit{rootPlace}, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{numActive} = @ + 2, \\
&\qquad\qquad\qquad\qquad\qquad ![\mathit{fid}].\mathit{transit}[\mathit{src}][\mathit{target}] = @ + 1,
\end{aligned}
$$

```
                                                    ![fid].live[rootPlace] = @ + 1]  root finish's task
          ELSE   p0fstates' = [p0fstates EXCEPT ![fid].numActive = @ + 1,
                                                ![fid].transit[src][target] = @ + 1]
    ELSE  AddTransitAdopted(fid, pfid, root, rootPlace, src, target)

TransitToLive(fid, src, target) ≜
  IF p0fstates[fid].adopterId = NotID
   THEN  ∧ p0fstates[fid].id ≠ NotID
         ∧ p0fstates[fid].transit[src][target] > 0
         ∧ p0fstates' = [p0fstates EXCEPT ![fid].transit[src][target] = @ − 1,
                                          ![fid].live[target] = @ + 1]
   ELSE  LET adopter ≜ p0fstates[fid].adopterId
         IN  ∧ p0fstates[adopter].transitAdopted[src][target] > 0
             ∧ p0fstates' = [p0fstates EXCEPT ![adopter].transitAdopted[src][target] = @ − 1,
                                              ![adopter].liveAdopted[target] = @ + 1]

LiveToCompleted(fid, target) ≜
  IF p0fstates[fid].adopterId = NotID
   THEN ∧ p0fstates[fid].numActive > 0
        ∧ p0fstates[fid].live[target]     > 0
        ∧ IF p0fstates[fid].numActive = 1
          THEN p0fstates' = [p0fstates EXCEPT ![fid].live[target] = @ − 1,
                                              ![fid].numActive = @ − 1,
                                              ![fid].isReleased = TRUE]
          ELSE p0fstates' = [p0fstates EXCEPT ![fid].live[target] = @ − 1,
                                              ![fid].numActive = @ − 1]
   ELSE  LET adopter ≜ p0fstates[fid].adopterId
         IN  ∧ p0fstates[adopter].liveAdopted[target] > 0
             ∧ p0fstates[adopter].numActive > 0
             ∧ p0fstates' = [p0fstates EXCEPT ![adopter].liveAdopted[target] = @ − 1,
                                              ![adopter].numActive = @ − 1]

Quiescent(fid) ≜
  IF p0fstates[fid].adopterId = NotID
   THEN p0fstates[fid].numActive = 1
   ELSE LET adopter ≜ p0fstates[fid].adopterId
        IN  p0fstates[adopter].numActive = 1

ReleaseFinishMsg(fid, here) ≜
  IF p0fstates[fid].adopterId = NotID
   THEN [mid  ↦ mseq,
         src  ↦ here,
         dst  ↦ p0fstates[fid].gfsRootPlace,
         fid  ↦ p0fstates[fid].gfsRoot,
         type ↦ "releaseFinish"]
   ELSE LET adopter ≜ p0fstates[fid].adopterId
```

```
        IN   [mid  ↦ mseq,
              src  ↦ here,
              dst  ↦ p0fstates[adopter].gfsRootPlace,
              fid  ↦ p0fstates[adopter].gfsRoot,
              type ↦ "releaseFinish"]

GetLostFIDs(dead) ≜
  {m ∈ Adopter : ∧ m.child ≠ NotID
                 ∧ m.adopter ≠ NotID
                 ∧ p0fstates[m.child].id ≠ NotID
                 ∧ p0fstates[m.child].adopterId = NotID
                 ∧ fstates[m.adopter].id ≠ NotID
                 ∧ fstates[m.child].here  = dead
                 ∧ m.adopter = GetAdopter(m.child, dead)
                 ∧ m.a = LiveAncestors(m.child, dead)
  }

GetAdoptionSeeker ≜
   IF p0adoptSet      = {} THEN NotAdoptor
    ELSE  CHOOSE m ∈ p0adoptSet : TRUE

GetConvertTasks ≜
  {t ∈ ConvTask :  ∧ t.pl ≠ NotPlace
                   ∧ t.fid ≠ NotID
                   ∧ p0fstates[t.fid].id ≠ NotID
                   ∧ p0fstates[t.fid].adopterId = NotID
                   ∧ isDead[PROG_HOME][t.pl] = FALSE
  }

GetConvertSeeker ≜
   IF p0convSet = {} THEN NotConvTask
    ELSE  CHOOSE m ∈ p0convSet : TRUE

CreateReleaseMessages ≜
  {m ∈ ReleaseFinishMessages : ∧ m.mid = mseq
                               ∧ m.src  = PROG_HOME
                               ∧ ∃ r ∈ IDRange : ∧ m.fid = r
                                                 ∧ p0fstates[r].id = r
                                                 ∧ p0fstates[r].numActive = 0
                                                 ∧ p0fstates[r].isReleased = FALSE
                                                 ∧ p0fstates[r].adopterId = NotID
                                                 ∧ m.dst = fstates[r].here
  }

├──────────────────────────────────────────────────────────────────────────────┤

RecvTransit(here) ≜
  ∧ p0state = "running"
```

$\land\ pstate = \text{“running”}$
$\land\ msgs \neq \{\}$
$\land$ LET $msg \triangleq FindIncomingMSG(here, \text{“transit”})$
$mid \triangleq msg.mid$
$fid \triangleq msg.fid$
$pfid \triangleq msg.pfid$
$root \triangleq msg.rfid$
$rootPlace \triangleq msg.rpl$
$src \triangleq msg.src$
$target \triangleq msg.target$
IN $\land\ src \neq NotPlace$
$\land\ fid \neq NotID$
$\land\ msg \neq NotMessage$
$\land$ IF $isDead[here][src]$
THEN $p0fstates' = p0fstates$
ELSE IF $isDead[here][target]$
THEN $AddException(fid, pfid, root, rootPlace, src, target)$
ELSE $AddTransit(fid, pfid, root, rootPlace, src, target)$
$\land\ ReplaceMsg([\ mid \mapsto mid,$
$src \mapsto src,$
$dst \mapsto here,$
$target \mapsto target,$
$fid \mapsto fid,$
$pfid \mapsto pfid,$
$rfid \mapsto root,$
$rpl \mapsto rootPlace,$
$type \mapsto \text{“transit”}],$
$[\ mid \mapsto mseq,$
$src \mapsto here,$
$dst \mapsto src,$
$fid \mapsto fid,$
$type \mapsto \text{“transitDone”}])$
$\land\ mseq' = mseq + 1$
$\land$ UNCHANGED $\langle fstates, pstate, program, aseq, fseq, p0dead, p0convSet,$
$readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar, p0adoptSet, p0state,$
$killed, killedCnt, pendingAct, isDead\rangle$

$RecvLive(here) \triangleq$
$\land\ p0state = \text{“running”}$
$\land\ pstate = \text{“running”}$
$\land\ msgs \neq \{\}$
$\land$ LET $msg \triangleq FindIncomingMSG(here, \text{“live”})$
$mid \triangleq msg.mid$
$fid \triangleq msg.fid$
$src \triangleq msg.src$

$target \triangleq msg.target$
$actId \triangleq msg.aid$
$submit \triangleq$ IF $isDead[here][src] \lor isDead[here][target]$
THEN FALSE
ELSE TRUE
IN $\land msg \neq NotMessage$
$\land$ IF $submit$
THEN $TransitToLive(fid, src, target)$
ELSE $p0fstates' = p0fstates$
$\land ReplaceMsg([mid \mapsto mid,$
$src \mapsto src,$
$dst \mapsto here,$
$target \mapsto target,$
$fid \mapsto fid,$
$aid \mapsto actId,$
$type \mapsto$ "live"$],$
$[mid \mapsto mseq,$
$src \mapsto here,$
$dst \mapsto target,$
$aid \mapsto actId,$
$submit \mapsto submit,$
$type \mapsto$ "liveDone"$])$
$\land mseq' = mseq + 1$
$\land$ UNCHANGED $\langle fstates, pstate, program, aseq, fseq, p0dead, p0convSet,$
$readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar, p0adoptSet, p0state,$
$killed, killedCnt, pendingAct, isDead\rangle$

$RecvCompleted(here) \triangleq$
$\land p0state =$ "running"
$\land pstate =$ "running"
$\land msgs \neq \{\}$
$\land$ LET $msg \triangleq FindIncomingMSG(here,$ "completed"$)$
$mid \triangleq msg.mid$
$fid \triangleq msg.fid$
$src \triangleq msg.src$
$target \triangleq msg.target$
IN $\land msg \neq NotMessage$
$\land$ IF $\neg isDead[here][target]$
THEN $LiveToCompleted(fid, target)$
ELSE $p0fstates' = p0fstates$
$\land$ IF $\land Quiescent(fid)$
$\land \neg isDead[here][target]$
THEN $\land ReplaceMsg([mid \mapsto mid,$
$src \mapsto src,$
$dst \mapsto here,$

```
                                        target ↦ target,
                                           fid ↦ fid,
                                          type ↦ "completed"],
                                        ReleaseFinishMsg(fid, here))
                        ∧ mseq′ = mseq + 1
              ELSE      ∧ RecvMsg([ mid ↦ mid,
                                    src ↦ src,
                                    dst ↦ here,
                                 target ↦ target,
                                    fid ↦ fid,
                                   type ↦ "completed"])
                        ∧ mseq′ = mseq
    ∧ UNCHANGED ⟨fstates, pstate, program, aseq, fseq, p0dead, p0convSet,
                 readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar, p0adoptSet, p0state,
                 killed, killedCnt, pendingAct, isDead⟩

SeekAdoption(here) ≜
  ∧ p0state = "seekAdoption"
  ∧ LET pair ≜ GetAdoptionSeeker
        lost ≜ pair.child
        adopter ≜ pair.adopter
    IN  IF pair = NotAdoptor
        THEN ∧ p0state′ = "convertDead"
             ∧ p0fstates′ = p0fstates
             ∧ p0adoptSet′ = p0adoptSet
             ∧ p0convSet′ = GetConvertTasks
        ELSE ∧ p0state′ = p0state
             ∧ p0fstates′ = [p0fstates EXCEPT ![lost].adopterId = adopter,
                                              ![adopter].liveAdopted = [p ∈ PLACE ↦   p0fstates[adop
                                                                                    + p0fstates[los
                                                                                    + p0fstates[los
                                                                                  ],
                                              ![adopter].transitAdopted = [p ∈ PLACE ↦
                                                                             [q ∈ PLACE ↦  p0fstates[a
                                                                                         + p0fstates[
                                                                                         + p0fstates[
                                                                           ]],
                                              ![adopter].numActive = @ + p0fstates[lost].numActive
                                ]
             ∧ p0adoptSet′ = p0adoptSet \ {pair}
             ∧ p0convSet′ = p0convSet
  ∧ UNCHANGED ⟨fstates, msgs, pstate, program, aseq, fseq, mseq, p0dead,
               readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar,
               killed, killedCnt, pendingAct, isDead⟩
```

$ConvertDeadActivities(here) \triangleq$
$\land\ p0state = \text{“convertDead”}$
$\land$ LET $t \triangleq GetConvertSeeker$
$pl \triangleq t.pl$
$fid \triangleq t.fid$
IN IF $fid = NotID$
THEN $\land\ p0convSet' = p0convSet$
$\land\ p0state' = \text{“release”}$
$\land\ p0fstates' = p0fstates$
ELSE $\land\ p0convSet' = p0convSet \setminus \{t\}$
$\land\ p0fstates' = [p0fstates$ EXCEPT $![fid].numActive = @ - p0fstates[fid].transit[pl][p0dea$
$- p0fstates[fid].transit[p0dead][p$
$- p0fstates[fid].transitAdopted[p$
$- p0fstates[fid].transitAdopted[p$
$- p0fstates[fid].live[p0dead]$
$- p0fstates[fid].liveAdopted[p0de$
$![fid].transit[pl][p0dead] = 0,$
$![fid].transitAdopted[pl][p0dead] = 0,$
$![fid].transit[p0dead][pl] = 0,$
$![fid].transitAdopted[p0dead][pl] = 0,$
$![fid].live[p0dead] = 0,$
$![fid].liveAdopted[p0dead] = 0$
]
$\land\ p0state' = p0state$
$\land$ UNCHANGED $\langle fstates, msgs, pstate, program, aseq, fseq, mseq, p0dead,$
$readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar,$
$killed, killedCnt, pendingAct, isDead, p0adoptSet\rangle$

$ReleaseAll(here) \triangleq$
$\land\ p0state = \text{“release”}$
$\land\ p0fstates' = [r \in IDRange \mapsto$ IF $p0fstates[r].numActive = 0 \land p0fstates[r].isReleased = \text{FALSE} \land p0fstate$
THEN [$id \mapsto p0fstates[r].id,$
$parent \mapsto p0fstates[r].parent,$
$gfsRoot \mapsto p0fstates[r].gfsRoot,$
$gfsRootPlace \mapsto p0fstates[r].gfsRootPlace,$
$numActive \mapsto p0fstates[r].numActive,$
$live \mapsto p0fstates[r].live,$
$transit \mapsto p0fstates[r].transit,$
$liveAdopted \mapsto p0fstates[r].liveAdopted,$
$transitAdopted \mapsto p0fstates[r].transitAdopted,$
$excs \mapsto p0fstates[r].excs,$
$adopterId \mapsto p0fstates[r].adopterId,$
$isReleased \mapsto \text{TRUE}$
]
ELSE $p0fstates[r]]$

$\land\ msgs' = msgs \cup CreateReleaseMessages$
$\land\ mseq' = mseq + 1$
$\land\ p0state' =$ "running"
$\land$ UNCHANGED $\langle fstates, pstate, program, aseq, fseq, p0dead,$
$readyQ, thrds, ppProgram, ppcurStmt, incPar, decPar, p0convSet,$
$killed, killedCnt, pendingAct, isDead, p0adoptSet\rangle$

$P0NotifyPlaceDeath(dead) \triangleq$
$\land\ p0state =$ "running"
$\land\ p0dead' = dead$
$\land\ p0adoptSet' = GetLostFIDs(dead)$
$\land\ p0state' =$ "seekAdoption"

* Modification History
* Last modified *Mon Nov* 06 19:10:23 *AEDT* 2017 by *u5482878*
* Last modified *Mon Nov* 06 01:15:25 *AEDT* 2017 by *shamouda*
* Created *Fri Oct* 13 15:15:59 *AEDT* 2017 by *u5482878*