MODULE $2PC$

EXTENDS *Integers*, *Sequences*, *FiniteSets*, *TLC*

CONSTANT RM, The set of participating resource managers $RM = 1 \,..\, 3$
$RMMAYFAIL$,
$BYZRM$,
$TMMAYFAIL$ Whether TM may fail $MAYFAIL =$ TRUE or FALSE

```
**************************************************************************
--algorithm TransactionCommit{
  variable rmState = [rm ∈ RM ↦ "working"],
           tmState = "init" ;
  define {
    canCommit ≜  ∀ rmc ∈ RM : rmState[rmc] ∈ {"prepared"}
                 ∨ ∃ rm ∈ RM : rmState[rm] ∈ {"committed"}  for when BTM takes over
    canAbort ≜  ∃ rm ∈ RM : rmState[rm] ∈ {"aborted", "failed"}
                ∧ ¬∃ rmc ∈ RM : rmState[rmc] = "committed"   inconsistent if commented
    }

  macro Prepare( p ) {
    await rmState[p] = "working" ;
    rmState[p] := "prepared" ;  }

  macro Decide( p ) {
    either { await tmState = "commit" ;
             rmState[p] := "committed" ;  }
               if (BYZRM) either rmState[p] := "committed" or rmState[p] := "aborted";}
    or     { await rmState[p] = "working" ∨ tmState = "abort" ;
             rmState[p] := "aborted" ;  }
    }

  macro Fail( p ) {
    if ( RMMAYFAIL ) rmState[p] := "failed" ;
    }

  fair  process ( RManager ∈ RM ) {
   RS: while ( rmState[self] ∈ {"working", "prepared"} ) {
         either Prepare(self) or Decide(self) or Fail(self) }
    }

  fair process ( TManager = 0 ) {
  TS: either { await canCommit ;
         TC:  tmState := "commit" ;
         F1:  if ( TMMAYFAIL ) tmState := "hidden" ;  }

      or { await canAbort ;
        TA: tmState := "abort" ;
        F2: if ( TMMAYFAIL ) tmState := "hidden" ;  }
```

```
  }
}
```

****

BEGIN TRANSLATION ($chksum(pcal) =$ "$a382282b$" $\land\ chksum(tla) =$ "$a9646dee$")

VARIABLES $rmState$, $tmState$, pc

define statement

$$
\begin{aligned}
canCommit \triangleq\ & \forall\, rmc \in RM : rmState[rmc] \in \{\text{“prepared”}\} \\
\lor\ & \exists\, rm \in RM : rmState[rm] \in \{\text{“committed”}\} \\
canAbort \triangleq\ & \exists\, rm \in RM : rmState[rm] \in \{\text{“aborted”}, \text{“failed”}\} \\
\land\ & \neg\exists\, rmc \in RM : rmState[rmc] = \text{“committed”}
\end{aligned}
$$

$$vars \triangleq \langle rmState,\ tmState,\ pc\rangle$$

$$ProcSet \triangleq (RM) \cup \{0\}$$

$Init \triangleq$ Global variables

$$
\begin{aligned}
& \land\ rmState = [rm \in RM \mapsto \text{“working”}] \\
& \land\ tmState = \text{“init”} \\
& \land\ pc = [self \in ProcSet \mapsto \text{CASE } self \in RM \to \text{“RS”} \\
& \qquad\qquad\qquad\qquad\qquad\qquad \square\ \ self = 0 \to \text{“TS”}]
\end{aligned}
$$

$$
\begin{aligned}
RS(self) \triangleq\ & \land\ pc[self] = \text{“RS”} \\
& \land \text{IF } rmState[self] \in \{\text{“working”}, \text{“prepared”}\} \\
& \quad \text{THEN } \land\ \lor\ \land\ rmState[self] = \text{“working”} \\
& \qquad\qquad\qquad\ \land\ rmState' = [rmState \text{ EXCEPT } ![self] = \text{“prepared”}] \\
& \qquad\qquad\ \lor\ \land\ \lor\ \land\ tmState = \text{“commit”} \\
& \qquad\qquad\qquad\qquad\ \land\ rmState' = [rmState \text{ EXCEPT } ![self] = \text{“committed”}] \\
& \qquad\qquad\qquad\ \lor\ \land\ rmState[self] = \text{“working”} \lor tmState = \text{“abort”} \\
& \qquad\qquad\qquad\qquad\ \land\ rmState' = [rmState \text{ EXCEPT } ![self] = \text{“aborted”}] \\
& \qquad\qquad\ \lor\ \land \text{IF } RMMAYFAIL \\
& \qquad\qquad\qquad\qquad \text{THEN } \land\ rmState' = [rmState \text{ EXCEPT } ![self] = \text{“failed”}] \\
& \qquad\qquad\qquad\qquad \text{ELSE } \land \text{TRUE} \\
& \qquad\qquad\qquad\qquad\qquad\ \land \text{UNCHANGED } rmState \\
& \qquad\ \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“RS”}] \\
& \quad \text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“Done”}] \\
& \qquad\ \land \text{UNCHANGED } rmState \\
& \land \text{UNCHANGED } tmState
\end{aligned}
$$

$$RManager(self) \triangleq RS(self)$$

$$
\begin{aligned}
TS \triangleq\ & \land\ pc[0] = \text{“TS”} \\
& \land\ \lor\ \land\ canCommit \\
& \qquad\ \land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“TC”}] \\
& \quad\ \lor\ \land\ canAbort
\end{aligned}
$$

$\land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“TA”}]$
$\land$ UNCHANGED $\langle rmState,\ tmState\rangle$

$TC \triangleq \land\ pc[0] = \text{“TC”}$
$\land\ tmState' = \text{“commit”}$
$\land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“F1”}]$
$\land$ UNCHANGED $rmState$

$F1 \triangleq \land\ pc[0] = \text{“F1”}$
$\land$ IF $TMMAYFAIL$
THEN $\land\ tmState' = \text{“hidden”}$
ELSE $\land$ TRUE
$\land$ UNCHANGED $tmState$
$\land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“Done”}]$
$\land$ UNCHANGED $rmState$

$TA \triangleq \land\ pc[0] = \text{“TA”}$
$\land\ tmState' = \text{“abort”}$
$\land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“F2”}]$
$\land$ UNCHANGED $rmState$

$F2 \triangleq \land\ pc[0] = \text{“F2”}$
$\land$ IF $TMMAYFAIL$
THEN $\land\ tmState' = \text{“hidden”}$
ELSE $\land$ TRUE
$\land$ UNCHANGED $tmState$
$\land\ pc' = [pc \text{ EXCEPT } ![0] = \text{“Done”}]$
$\land$ UNCHANGED $rmState$

$TManager \triangleq TS \lor TC \lor F1 \lor TA \lor F2$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq \land \forall\, self \in ProcSet : pc[self] = \text{“Done”}$
$\land$ UNCHANGED $vars$

$Next \triangleq TManager$
$\lor (\exists\, self \in RM : RManager(self))$
$\lor Terminating$

$Spec \triangleq \land\ Init \land \Box[Next]_{vars}$
$\land \forall\, self \in RM : \mathrm{WF}_{vars}(RManager(self))$
$\land \mathrm{WF}_{vars}(TManager)$

$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$

END TRANSLATION

\ * Modification History
\ * Last modified Sat May 15 23:25:58 *CDT* 2021 by *sridhar*
\ * Created Sat May 15 23:23:14 *CDT* 2021 by *sridhar*