MODULE *ACP_NB*

Time-stamp: < 10 *Jun* 2002 at 14:06:57 by charpov on *berlioz.cs.unh.edu* >

Non blocking Atomic *Committment* Protocol (ACP-NB)
The non blocking property $AC5$ is obtained by using a reliable broadcast
implemented as follows:
- upon reception of a broadcast message, this message is forwarded to all participants before it's delivered to the local site;
- since participant $i$ does not forward to itself, $forward[i]$ is used to store the decision before it's delivered (and becomes "decision")

EXTENDS *ACP_SB*

---

Participants type is extended with a "forward" variable.
Coordinator type is unchanged.

$$
\begin{array}{l}
TypeInvParticipantNB \triangleq participant \in [ \\
\quad participants \to [ \\
\quad\quad vote : \{yes, no\}, \\
\quad\quad alive : \text{BOOLEAN}, \\
\quad\quad decision : \{undecided, commit, abort\}, \\
\quad\quad faulty : \text{BOOLEAN}, \\
\quad\quad voteSent : \text{BOOLEAN}, \\
\quad\quad forward : [participants \to \{notsent, commit, abort\}] \\
\quad ] \\
]
\end{array}
$$

$$TypeInvNB \triangleq TypeInvParticipantNB \land TypeInvCoordinator$$

---

Initially, participants have not forwarded anything yet

$$
\begin{array}{l}
InitParticipantNB \triangleq participant \in [ \\
\quad participants \to [ \\
\quad\quad vote : \{yes, no\}, \\
\quad\quad alive : \{\text{TRUE}\}, \\
\quad\quad decision : \{undecided\}, \\
\quad\quad faulty : \{\text{FALSE}\}, \\
\quad\quad voteSent : \{\text{FALSE}\}, \\
\quad\quad forward : [participants \to \{notsent\}] \\
\quad ] \\
]
\end{array}
$$

$$InitNB \triangleq InitParticipantNB \land InitCoordinator$$

---

Participant statements that realize a better broadcast

$forward(i, j)$: forwarding of the predecision from participant $i$ to participant $j$
IF
participant $i$ is alive
participant $i$ has received a decision (stored in $forward[i]$)
participant $i$ has not yet forwarded this decision to participant $j$
THEN
participant $i$ forwards the decision to participant $j$

$$
\begin{array}{l}
forward(i, j) \triangleq \land i \neq j \\
\qquad\qquad \land participant[i].alive \\
\qquad\qquad \land participant[i].forward[i] \neq notsent \\
\qquad\qquad \land participant[i].forward[j] = notsent \\
\qquad\qquad \land participant' = [participant \text{ EXCEPT } ![i] =
\end{array}
$$

```
                    [@ EXCEPT !.forward =
                      [@ EXCEPT ![j] = participant[i].forward[i]]
                    ]
                  ]
                ∧ UNCHANGED ⟨coordinator⟩
```

*preDecideOnForward*$(i, j)$: participant $i$ receives decision from participant $j$
IF
participant $i$ is alive
participant $i$ has yet to receive a decision
participant $j$ has forwarded its decision to participant $i$
THEN
participant $i$ (pre)decides in accordance with participant $j$'s decision

```
preDecideOnForward(i, j) ≜ ∧ i ≠ j
                           ∧ participant[i].alive
                           ∧ participant[i].forward[i] = notsent
                           ∧ participant[j].forward[i] ≠ notsent
                           ∧ participant' = [participant EXCEPT ![i] =
                               [@ EXCEPT !.forward =
                                 [@ EXCEPT ![i] = participant[j].forward[i]]
                               ]
                             ]
                           ∧ UNCHANGED ⟨coordinator⟩
```

*preDecide*$(i)$: participant $i$ receives decision from coordinator
IF
participant $i$ is alive
participant $i$ has yet to receive a decision
coordinator has sent its decision to participant $i$
THEN
participant $i$ (pre)decides in accordance with coordinator's decision

```
preDecide(i) ≜ ∧ participant[i].alive
               ∧ participant[i].forward[i] = notsent
               ∧ coordinator.broadcast[i] ≠ notsent
               ∧ participant' = [participant EXCEPT ![i] =
                   [@ EXCEPT !.forward =
                     [@ EXCEPT ![i] = coordinator.broadcast[i]]
                   ]
                 ]
               ∧ UNCHANGED ⟨coordinator⟩
```

*decideNB*$(i)$: Actual decision, after predecision has been forwarded
IF
participant $i$ is alive
participant $i$ has forwarded its (pre)decision to all other participants
THEN
participant $i$ decides in accordance with it's predecision

```
decideNB(i) ≜ ∧ participant[i].alive
              ∧ ∀ j ∈ participants : participant[i].forward[j] ≠ notsent
              ∧ participant' = [participant EXCEPT ![i] =
                  [@ EXCEPT !.decision = participant[i].forward[i]]
                ]
              ∧ UNCHANGED ⟨coordinator⟩
```

*abortOnTimeout*$(i)$: conditions for a timeout are simulated
IF
participant is alive and undecided and coordinator is not alive

coordinator died before sending decision to all participants who are alive
all dead participants died before forwarding decision to a participant who is alive
THEN
decide abort

$$
\begin{aligned}
abortOnTimeout(i) \triangleq\ & \land\ participant[i].alive \\
& \land\ participant[i].decision = undecided \\
& \land\ \neg coordinator.alive \\
& \land\ \forall\, j \in participants : participant[j].alive \Rightarrow coordinator.broadcast[j] = notsent \\
& \land\ \forall\, j,\, k \in participants : \neg participant[j].alive \land participant[k].alive \Rightarrow participant[j].forward[k] = notsent \\
& \land\ participant' = [participant \text{ EXCEPT } ![i] = [@ \text{ EXCEPT } !.decision = abort]] \\
& \land\ \text{UNCHANGED } \langle coordinator \rangle
\end{aligned}
$$

---

FOR $N$ PARTICIPANTS

$$
\begin{aligned}
parProgNB(i,\, j) \triangleq\ & \lor\ sendVote(i) \\
& \lor\ abortOnVote(i) \\
& \lor\ abortOnTimeoutRequest(i) \\
& \lor\ forward(i,\, j) \\
& \lor\ preDecideOnForward(i,\, j) \\
& \lor\ abortOnTimeout(i) \\
& \lor\ preDecide(i) \\
& \lor\ decideNB(i)
\end{aligned}
$$

$$parProgNNB \triangleq \exists\, i,\, j \in participants : parDie(i) \lor parProgNB(i,\, j)$$

$$progNNB \triangleq parProgNNB \lor coordProgN$$

$$
\begin{aligned}
fairnessNB \triangleq\ & \land\ \forall\, i \in participants : \mathrm{WF}_{\langle coordinator,\, participant \rangle}(\exists\, j \in participants : parProgNB(i,\, j)) \\
& \land\ \mathrm{WF}_{\langle coordinator,\, participant \rangle}(coordProgB)
\end{aligned}
$$

$$SpecNB \triangleq InitNB \land \Box[progNNB]_{\langle coordinator,\, participant \rangle} \land fairnessNB$$

---

(SOME) INVALID PROPERTIES

$$AllCommit \triangleq \forall\, i \in participants : \Diamond(participant[i].decision = commit \lor participant[i].faulty)$$

$$AllAbort \triangleq \forall\, i \in participants : \Diamond(participant[i].decision = abort \lor participant[i].faulty)$$

$$
\begin{aligned}
AllCommitYesVotes \triangleq\ & \forall\, i \in participants : \\
& \quad \forall\, j \in participants : participant[j].vote = yes \\
& \rightsquigarrow participant[i].decision = commit \lor participant[i].faulty \lor coordinator.faulty
\end{aligned}
$$

---