MODULE *OpAWSet*

EXTENDS *AWSet*
CONSTANTS *Read*(_)

VARIABLES

aset,	*aset*[*r*]: the set of active elements maintained by $r \in Replica$
abuf,	*abuf*[*r*]: the buffer for elements added by $r \in Replica$ since the last broadcast
rbuf,	*rbuf*[*r*]: the buffer for elements removed by $r \in Replica$ since the last broadcast
variables for network:	
incoming,	*incoming*[*r*]: incoming channel at replica $r \in Replica$
lmsg,	*lmsg*[*r*]: the last message delivered at $r \in Replica$ to the upper-layer protocol
vc,	*vc*[*r*][*s*] denotes the latest message from $s \in Replica$ observed by $r \in Replica$
variables for *SEC*:	
uset,	*uset*[*r*]: the set of updates seen by replica $r \in Replica$
uincoming,	*uincoming*[*r*]: incoming channel for broadcasting/delivering updates at $r \in Replica$
buset	*buset*[*r*]: the buffer of local updates made by $r \in Replica$ since the last broadcast

$nVars \triangleq \langle incoming, lmsg, vc \rangle$
$secVars \triangleq \langle uset, uincoming, buset \rangle$
$vars \triangleq \langle aset, abuf, rbuf, seq, nVars, secVars \rangle$

$Msg \triangleq [aid : Aid, abuf : \text{SUBSET } Element, rbuf : \text{SUBSET } Element]$
$Network \triangleq \text{INSTANCE } ReliableCausalNetwork \text{ WITH } incoming \leftarrow incoming, lmsg \leftarrow lmsg, vc \leftarrow vc$

$ReadOpAWSet(r) \triangleq \{ele.d : ele \in aset[r]\}$ read the state of $r \in Replica$
$SEC \triangleq \text{INSTANCE } OpSEC \text{ WITH } uset \leftarrow uset, uincoming \leftarrow uincoming, buset \leftarrow buset$

$TypeOK \triangleq$
- $\land\ aset \in [Replica \to \text{SUBSET } Element]$
- $\land\ abuf \in [Replica \to \text{SUBSET } Element]$
- $\land\ rbuf \in [Replica \to \text{SUBSET } Element]$
- $\land\ IntTypeOK$
- $\land\ Network!SMTypeOK$
- $\land\ SEC!SECTypeOK$

$Init \triangleq$
- $\land\ aset = [r \in Replica \mapsto \{\}]$
- $\land\ abuf = [r \in Replica \mapsto \{\}]$
- $\land\ rbuf = [r \in Replica \mapsto \{\}]$
- $\land\ IntInit$
- $\land\ Network!RCNInit$
- $\land\ SEC!OpSECInit$

1
1
1

$$Add(d, r) \triangleq \quad r \in Replica \text{ adds } d \in Data$$

$$\begin{aligned}
&\land \text{LET } e \triangleq [aid \mapsto [r \mapsto r,\ seq \mapsto seq[r]],\ d \mapsto d] \\
&\quad \text{IN } \land aset' = [aset \text{ EXCEPT } ![r] = @ \cup \{e\}] \\
&\qquad\ \ \land abuf' = [abuf \text{ EXCEPT } ![r] = @ \cup \{e\}] \\
&\land IntDo(r) \\
&\land SEC!OpSECDo(r) \\
&\land \text{UNCHANGED } \langle rbuf,\ nVars \rangle
\end{aligned}$$

$$Remove(d, r) \triangleq \quad r \in Replica \text{ removes } d \in Data$$

$$\begin{aligned}
&\land \text{LET } E \triangleq \{ele \in aset[r] : ele.d = d\} \quad E \text{ may be empty} \\
&\quad \text{IN } \land aset' = [aset \text{ EXCEPT } ![r] = @ \setminus E] \\
&\qquad\ \ \land rbuf' = [rbuf \text{ EXCEPT } ![r] = @ \cup E] \\
&\land IntDo(r) \\
&\land SEC!OpSECDo(r) \\
&\land \text{UNCHANGED } \langle abuf,\ nVars \rangle
\end{aligned}$$

$$Do(r) \triangleq \quad \text{We ignore } ReadOpAWSet(r) \text{ since it does not modify states.}$$

$$\exists\, d \in Data : Add(d, r) \lor Remove(d, r)$$

$$Send(r) \triangleq \quad r \in Replica \text{ sends a message}$$

$$\begin{aligned}
&\land \lor abuf[r] \neq \{\} \\
&\quad\ \lor rbuf[r] \neq \{\} \\
&\land abuf' = [abuf \text{ EXCEPT } ![r] = \{\}] \\
&\land rbuf' = [rbuf \text{ EXCEPT } ![r] = \{\}] \\
&\land Network!RCNBroadcast(r, [aid \mapsto [r \mapsto r,\ seq \mapsto seq[r]], \\
&\qquad\qquad\qquad\qquad\qquad\quad abuf \mapsto abuf[r],\ rbuf \mapsto rbuf[r]]) \\
&\land IntSend(r) \\
&\land SEC!OpSECSend(r) \\
&\land \text{UNCHANGED } \langle aset \rangle
\end{aligned}$$

$$Deliver(r) \triangleq \quad r \in Replica \text{ delivers a message } (lmsg'[r])$$

$$\begin{aligned}
&\land IntDeliver(r) \\
&\land Network!RCNDeliver(r) \\
&\land SEC!OpSECDeliver(r,\ lmsg'[r].aid) \\
&\land aset' = [aset \text{ EXCEPT } ![r] = (@ \cup lmsg'[r].abuf) \setminus lmsg'[r].rbuf] \\
&\land \text{UNCHANGED } \langle abuf,\ rbuf \rangle
\end{aligned}$$

$$Next \triangleq \exists\, r \in Replica : Do(r) \lor Send(r) \lor Deliver(r)$$

$$Spec \triangleq Init \land \Box[Next]_{vars}$$

---

* Modification History
* Last modified *Wed Jun* 26 15:11:53 *CST* 2019 by *xhdn*
* Created *Fri* May 24 14:12:26 *CST* 2019 by *xhdn*