1 ────────────────── MODULE *AJupiterExtended* ──────────────────

AJupiter extended with *JupiterCtx*. This is used to show that *AJupiter* implements *XJupiter*.

5 EXTENDS *JupiterCtx*, *BufferStateSpace*   *TODO*: To extend *AJupiter*

6 ├───

7 VARIABLES *cbuf*, *crec*, *sbuf*, *srec*, *cincomingXJ*, *sincomingXJ*

8 $varsEx \triangleq \langle intVars, ctxVars, cbuf, crec, sbuf, srec, cincomingXJ, sincomingXJ \rangle$

10 $AJMsgEx \triangleq [ack : Nat, cop : Cop, oid : Oid]$

11 $commXJ \triangleq$ INSTANCE *CSComm* WITH $Msg \leftarrow Seq(Cop)$,

12 $cincoming \leftarrow cincomingXJ, sincoming \leftarrow sincomingXJ$

13 ├───

14 $TypeOKEx \triangleq$

15 $\land TypeOKInt$

16 $\land TypeOKCtx$

17 $\land commXJ!TypeOK$

18 $\land crec \in [Client \rightarrow Nat]$

19 $\land srec \in [Client \rightarrow Nat]$

20 $\land cbuf \in [Client \rightarrow Seq(Cop)]$

21 $\land sbuf \in [Client \rightarrow Seq(Cop)]$

22 ├───

23 $InitEx \triangleq$

24 $\land InitInt$

25 $\land InitCtx$

26 $\land commXJ!Init$

27 $\land crec = [c \in Client \mapsto 0]$

28 $\land srec = [c \in Client \mapsto 0]$

29 $\land cbuf = [c \in Client \mapsto \langle\rangle]$

30 $\land sbuf = [c \in Client \mapsto \langle\rangle]$

31 ├───

32 $DoOpEx(c, op) \triangleq$

33 LET $cop \triangleq [op \mapsto op, oid \mapsto [c \mapsto c, seq \mapsto cseq[c]], ctx \mapsto ds[c]]$

34 IN $\land crec' = [crec \text{ EXCEPT } ![c] = 0]$

35 $\land cbuf' = [cbuf \text{ EXCEPT } ![c] = Append(@, cop)]$

36 $\land SetNewAop(c, op)$

37 $\land Comm!CSend([ack \mapsto crec[c], cop \mapsto cop, oid \mapsto cop.oid])$

38 $\land commXJ!CSend(cop)$

40 $ClientPerformEx(c, m) \triangleq$

41 LET $xform \triangleq xFormShift(COT, m.cop, cbuf[c], m.ack)$

42 IN $\land cbuf' = [cbuf \text{ EXCEPT } ![c] = xform.xops]$

43 $\land crec' = [crec \text{ EXCEPT } ![c] = @ + 1]$

44 $\land SetNewAop(c, xform.xop.op)$

46 $ServerPerformEx(m) \triangleq$

47 LET $c \triangleq ClientOf(m.cop)$

1

```
48                xform  ≜  xFormShift(COT, m.cop, sbuf[c], m.ack)
49                 xcop  ≜  xform.xop
50         IN     ∧ srec' = [cl ∈ Client ↦ IF cl = c THEN srec[cl] + 1 ELSE  0]
51                ∧ sbuf' = [cl ∈ Client ↦ IF cl = c THEN xform.xops
52                                                     ELSE  Append(sbuf[cl], xcop)]
53                ∧ SetNewAop(Server, xcop.op)
54                ∧ Comm!SSend(c, [cl ∈ Client ↦
55                                      [ack ↦ srec[cl], cop ↦ xcop, oid ↦ xcop.oid]])
56                ∧ commXJ!SSendSame(c, xcop)
```
_____

```
58   DoEx(c)  ≜
59        ∧ DoInt(DoOpEx, c)
60        ∧ DoCtx(c)
61        ∧ UNCHANGED ⟨sbuf, srec⟩

63   RevEx(c)  ≜
64        ∧ RevInt(ClientPerformEx, c)
65        ∧ RevCtx(c)
66        ∧ commXJ!CRev(c)
67        ∧ UNCHANGED ⟨sbuf, srec⟩

69   SRevEx  ≜
70        ∧   SRevInt(ServerPerformEx)
71        ∧   SRevCtx
72        ∧   commXJ!SRev
73        ∧   UNCHANGED ⟨cbuf, crec⟩
```
_____

```
75   NextEx  ≜
76        ∨   ∃ c ∈ Client : DoEx(c) ∨ RevEx(c)
77        ∨   SRevEx

79   FairnessEx  ≜
80        WF_{varsEx}(SRevEx ∨ ∃ c ∈ Client : RevEx(c))

82   SpecEx  ≜  InitEx ∧ □[NextEx]_{varsEx}   ∧ FairnessEx
```
_____

```
84   QC  ≜    Quiescent Consistency
85        Comm!EmptyChannel ⇒ Cardinality(Range(state)) = 1

87   THEOREM  SpecEx ⇒ □QC
```
_____

* Modification History
* Last modified *Thu Jan 17 10:38:50 CST 2019* by *hengxin*
* Created *Thu Dec 27 21:15:09 CST 2018* by *hengxin*