```
1  ---------- MODULE AJupiterImplXJupiter ----------
   We show that AJupiter (specifically, AJupiterExtended) implements XJupiter.
5  EXTENDS AJupiterExtended, StateSpace
6  ----------
7  VARIABLES c2ss, s2ss

9  varsImpl ≜ ⟨varsEx, c2ss, s2ss⟩
10 ----------
11 TypeOKImpl ≜
12     ∧ TypeOKEx
13     ∧ ∀ c ∈ Client : IsSS(c2ss[c]) ∧ IsSS(s2ss[c])
14 ----------
15 InitImpl ≜
16     ∧ InitEx
17     ∧ c2ss = [c ∈ Client ↦ EmptySS]
18     ∧ s2ss = [c ∈ Client ↦ EmptySS]
19 ----------
   Client c ∈ Client issues an operation op.
23 DoOpImpl(c, op) ≜
24     LET   cop ≜ [op ↦ op, oid ↦ [c ↦ c, seq ↦ cseq'[c]], ctx ↦ ds[c]]
25     IN    ∧ crec' = [crec EXCEPT ![c] = 0]
26           ∧ cbuf' = [cbuf EXCEPT ![c] = Append(@, cop)]
27           ∧ state' = [state EXCEPT ![c] = Apply(op, @)]
28           ∧ Comm(Msg)!CSend([ack ↦ crec[c], cop ↦ cop, oid ↦ cop.oid])
29           ∧ commXJ!CSend(cop)
30           ∧ c2ss' = [c2ss EXCEPT ![c] =
31                           @ ⊕ [node ↦ {ds'[c]},
32                                edge ↦ {[from ↦ ds[c], to ↦ ds'[c], cop ↦ cop]}]
33                     ]
34           ∧ UNCHANGED s2ss

36 DoInsImpl(c) ≜
37     ∃ ins ∈ {op ∈ Ins : op.pos ∈ 1 .. (Len(state[c]) + 1) ∧ op.ch ∈ chins ∧ op.pr = Priority[c]} :
38        ∧ DoOpImpl(c, ins)
39        ∧ chins' = chins \ {ins.ch}

41 DoDelImpl(c) ≜
42     ∃ del ∈ {op ∈ Del : op.pos ∈ 1 .. Len(state[c])} :
43        ∧ DoOpImpl(c, del)
44        ∧ UNCHANGED chins

46 DoImpl(c) ≜
47     ∧ DoCtx(c)
48     ∧ ∨ DoInsImpl(c)
49       ∨ DoDelImpl(c)
50     ∧ UNCHANGED ⟨sbuf, srec⟩
```

```
RevImpl(c) ≜
    ∧  RevEx(c)
    ∧  LET m ≜ Head(cincoming[c])
           cBuf ≜ cbuf[c]
           cShiftedBuf ≜ SubSeq(cBuf, m.ack + 1, Len(cBuf))
           xform ≜ xFormCopCopsSS(m.cop, cShiftedBuf) [lss, xss]
       IN  c2ss' = [c2ss EXCEPT ![c] = @ ⊕ xform.xss]
    ∧  UNCHANGED s2ss
```

```
SRevImpl ≜
    ∧ SRevEx
    ∧ LET m ≜ Head(sincoming)
          c ≜ ClientOf(m.cop)
          cBuf ≜ sbuf[c]
          cShiftedBuf ≜ SubSeq(cBuf, m.ack + 1, Len(cBuf))
          xform ≜ xFormCopCopsSS(m.cop, cShiftedBuf) [lss, xss]
      IN  s2ss' = [cl ∈ Client ↦
                     IF cl = c THEN s2ss[cl] ⊕ xform.xss ELSE s2ss[cl] ⊕ xform.lss]
    ∧ UNCHANGED c2ss
```

```
NextImpl ≜
    ∨ ∃ c ∈ Client : DoImpl(c) ∨ RevImpl(c)
    ∨ SRevImpl

FairnessImpl ≜
    WF_varsImpl(SRevImpl ∨ ∃ c ∈ Client : RevImpl(c))

SpecImpl ≜ InitImpl ∧ □[NextImpl]_varsImpl  ∧ FairnessImpl

XJ ≜ INSTANCE XJupiter WITH
          cincoming ← cincomingXJ, sincoming ← sincomingXJ

THEOREM SpecImpl ⇒ XJ!Spec
```

* Modification History
* Last modified Sun *Dec* 30 16:48:09 *CST* 2018 by *hengxin*
* Created Sat *Dec* 29 18:36:51 *CST* 2018 by *hengxin*