MODULE *XJupiter*

Specification of the *Jupiter* protocol described in *CSCW'*2014 by *Xu*, Sun, and *Li*.

EXTENDS *GraphStateSpace*

VARIABLES

c2ss, *c2ss*[*c*]: the 2*D* state space (2*ss*, for short) at client $c \in Client$

s2ss *s2ss*[*c*]: the 2*D* state space maintained by the *Server* for client $c \in Client$

$vars \triangleq \langle intVars,\ ctxVars,\ c2ss,\ s2ss \rangle$

$TypeOK \triangleq$

$\land$ *TypeOKInt*

$\land$ *TypeOKCtx*

$\land$ $\forall\, c \in Client : IsSS(c2ss[c]) \land IsSS(s2ss[c])$

$Init \triangleq$

$\land\ InitInt$

$\land\ InitCtx$

$\land\ c2ss = [c \in Client \mapsto EmptySS]$

$\land\ s2ss = [c \in Client \mapsto EmptySS]$

$NextEdge(r,\ u,\ ss) \triangleq$ Return the unique outgoing edge from u in 2*D* state space *ss*

CHOOSE $e \in ss.edge : e.from = u$ before a transformation at u (r is not used).

$ClientPerform(c,\ cop) \triangleq$

LET $xform \triangleq xForm(NextEdge,\ c,\ cop,\ c2ss[c])$ *xform*: [*xcop*, *xss*, *lss*]

IN $\land\ c2ss' = [c2ss$ EXCEPT $![c] = @ \oplus xform.xss]$

$\land\ SetNewAop(c,\ xform.xcop.op)$

$ServerPerform(cop) \triangleq$

LET $c \triangleq ClientOf(cop)$

$xform \triangleq xForm(NextEdge,\ Server,\ cop,\ s2ss[c])$ *xform*: [*xcop*, *xss*, *lss*]

$xcop \triangleq xform.xcop$

IN $\land\ s2ss' = [cl \in Client \mapsto$ IF $cl = c$

THEN $s2ss[cl] \oplus xform.xss$

ELSE $s2ss[cl] \oplus xform.lss]$

$\land\ SetNewAop(Server,\ xcop.op)$

$\land\ Comm!SSendSame(c,\ xcop)$ broadcast the transformed *xcop*

$DoOp(c,\ op) \triangleq$

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

IN $\land\ ClientPerform(c,\ cop)$

$\land\ Comm!CSend(cop)$

$Do(c) \triangleq$

$\land\ DoInt(DoOp,\ c)$

$\land\ DoCtx(c)$
$\land$ UNCHANGED $s2ss$

$Rev(c) \triangleq$
$\land\ RevInt(ClientPerform,\ c)$
$\land\ RevCtx(c)$
$\land$ UNCHANGED $s2ss$

$SRev \triangleq$
$\land\ SRevInt(ServerPerform)$
$\land\ SRevCtx$
$\land$ UNCHANGED $c2ss$

$Next \triangleq$
$\lor \exists\, c \in Client : Do(c) \lor Rev(c)$
$\lor\ SRev$

$Fairness \triangleq$
$\mathrm{WF}_{vars}(SRev \lor \exists\, c \in Client : Rev(c))$

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

$CSSync \triangleq$ Each client $c \in Client$ is synchonized with the $Server$.
$\forall\, c \in Client : (ds[c] = ds[Server]) \Rightarrow c2ss[c] = s2ss[c]$

THEOREM $Spec \Rightarrow \Box CSSync$

---

* Modification History
* Last modified Sat *Jan* 12 15:57:05 *CST* 2019 by anonymous
* Created *Tue Oct* 09 16:33:18 *CST* 2018 by anonymous