

MODULE <i>AJupiter</i>	
Specification of the <i>Jupiter</i> protocol presented by Attiya et al.	
EXTENDS <i>JupiterInterface</i> , <i>OT</i> , <i>BufferStateSpace</i>	
VARIABLES	
<i>cbuf</i> ,	<i>cbuf</i> [<i>c</i>]: buffer for locally generated operations at client $c \in Client$
<i>crec</i> ,	<i>crec</i> [<i>c</i>]: number of remote operations received by client $c \in Client$ since the last time a local operation was generated
<i>sbuf</i> ,	<i>sbuf</i> [<i>c</i>]: buffer for transformed remote operations <i>w.r.t</i> client $c \in Client$
<i>srec</i>	<i>srec</i> [<i>c</i>]: number of locally generated operations by client $c \in Client$ since the last time a remote operation was transformed at the <i>Server</i>
$vars \triangleq \langle intVars, cbuf, crec, sbuf, srec \rangle$	
$AJMsg \triangleq$	
[<i>c</i> : <i>Client</i> , <i>ack</i> : <i>Nat</i> , <i>op</i> : <i>Op</i> \cup { <i>Nop</i> }] \cup messages sent to the <i>Server</i> from client $c \in Client$	
[<i>ack</i> : <i>Nat</i> , <i>op</i> : <i>Op</i> \cup { <i>Nop</i> }] messages broadcast to Clients from the <i>Server</i>	
$TypeOK \triangleq$	
$\wedge TypeOKInt$	
$\wedge cbuf \in [Client \rightarrow Seq(Op \cup \{Nop\})]$	
$\wedge crec \in [Client \rightarrow Nat]$	
$\wedge sbuf \in [Client \rightarrow Seq(Op \cup \{Nop\})]$	
$\wedge srec \in [Client \rightarrow Nat]$	
$Init \triangleq$	
$\wedge InitInt$	
$\wedge cbuf = [c \in Client \mapsto \langle \rangle]$	
$\wedge crec = [c \in Client \mapsto 0]$	
$\wedge sbuf = [c \in Client \mapsto \langle \rangle]$	
$\wedge srec = [c \in Client \mapsto 0]$	
$ClientPerform(c, m) \triangleq$	
LET $xform \triangleq xFormShift(OT, m.op, cbuf[c], m.ack)$ [<i>xop</i> , <i>xops</i>]	
IN $\wedge cbuf' = [cbuf \text{ EXCEPT } ![c] = xform.xops]$	
$\wedge crec' = [crec \text{ EXCEPT } ![c] = @ + 1]$	
$\wedge SetNewAop(c, xform.xop)$	
$ServerPerform(m) \triangleq$	
LET $c \triangleq m.c$	
$xform \triangleq xFormShift(OT, m.op, sbuf[c], m.ack)$ [<i>xop</i> , <i>xops</i>]	
$xop \triangleq xform.xop$	
IN $\wedge srec' = [cl \in Client \mapsto \text{IF } cl = c \text{ THEN } srec[cl] + 1 \text{ ELSE } 0]$	
$\wedge sbuf' = [cl \in Client \mapsto \text{IF } cl = c \text{ THEN } xform.xops$	
ELSE $Append(sbuf[cl], xop)]$	

$$\begin{array}{l}
\wedge \text{SetNewAop}(\text{Server}, \text{xop}) \\
\wedge \text{Comm!SSend}(c, [cl \in \text{Client} \mapsto [ack \mapsto \text{srec}[cl], op \mapsto \text{xop}]]) \\
\hline
\text{DoOp}(c, op) \triangleq \\
\quad \wedge \text{SetNewAop}(c, op) \\
\quad \wedge \text{cbuf}' = [\text{cbuf} \text{ EXCEPT } ![c] = \text{Append}(@, op)] \\
\quad \wedge \text{crec}' = [\text{crec} \text{ EXCEPT } ![c] = 0] \\
\quad \wedge \text{Comm!CSend}([c \mapsto c, ack \mapsto \text{crec}[c], op \mapsto op]) \\
\\
\text{Do}(c) \triangleq \\
\quad \wedge \text{DoInt}(\text{DoOp}, c) \\
\quad \wedge \text{UNCHANGED } \langle \text{sbuf}, \text{srec} \rangle \\
\\
\text{Rev}(c) \triangleq \\
\quad \wedge \text{RevInt}(\text{ClientPerform}, c) \\
\quad \wedge \text{UNCHANGED } \langle \text{sbuf}, \text{srec} \rangle \\
\\
\text{SRev} \triangleq \\
\quad \wedge \text{SRevInt}(\text{ServerPerform}) \\
\quad \wedge \text{UNCHANGED } \langle \text{cbuf}, \text{crec} \rangle \\
\hline
\text{Next} \triangleq \\
\quad \vee \exists c \in \text{Client} : \text{Do}(c) \vee \text{Rev}(c) \\
\quad \vee \text{SRev} \\
\\
\text{Fairness} \triangleq \\
\quad \text{WF}_{\text{vars}}(\text{SRev} \vee \exists c \in \text{Client} : \text{Rev}(c)) \\
\\
\text{Spec} \triangleq \text{Init} \wedge \Box[\text{Next}]_{\text{vars}} \quad \wedge \text{Fairness} \\
\hline
\text{QC} \triangleq \quad \text{Quiescent Consistency} \\
\quad \text{Comm!EmptyChannel} \Rightarrow \text{Cardinality}(\text{Range}(\text{state})) = 1 \\
\\
\text{THEOREM } \text{Spec} \Rightarrow \Box \text{QC} \\
\hline
\end{array}$$

\ * Modification History
\ * Last modified *Thu Jan 17 10:30:39 CST 2019* by anonymous
\ * Created *Satchins, Jun 23 17:14:18 CST 2018* by anonymous