MODULE *Config*

EXTENDS
Northbound,
Proposal,
Configuration,
Mastership,
Southbound

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

LOCAL INSTANCE *TLC*

$vars \triangleq \langle proposal, configuration, mastership, target \rangle$

Formal specification, constraints, and theorems.

$Init \triangleq$
$\land\ InitNorthbound$
$\land\ InitProposal$
$\land\ InitConfiguration$
$\land\ InitMastership$
$\land\ InitSouthbound$

$Next \triangleq$
$\lor\ \land\ NextNorthbound$
$\quad\land$ UNCHANGED $\langle\rangle$
$\lor\ \land\ NextProposal$
$\quad\land$ UNCHANGED $\langle\rangle$
$\lor\ \land\ NextConfiguration$
$\quad\land$ UNCHANGED $\langle proposal \rangle$
$\lor\ \land\ NextMastership$
$\quad\land$ UNCHANGED $\langle proposal, configuration \rangle$
$\lor\ \land\ NextSouthbound$
$\quad\land$ UNCHANGED $\langle proposal, configuration, mastership \rangle$

$Spec \triangleq Init \land \Box[Next]_{vars} \land \mathrm{WF}_{vars}(Next)$

$Order \triangleq$
$\forall\, i \in$ DOMAIN $proposal :$
$\quad\land\ proposal[i].type = ProposalChange$
$\quad\land\ proposal[i].phase = ProposalApply$
$\quad\land\ proposal[i].state\ \in \{ProposalComplete, ProposalFailed\}$

$\Rightarrow \forall\, j \in \text{DOMAIN } proposal :$
$j < i \Rightarrow$
$\land\ proposal[j].phase = ProposalCommit \Rightarrow proposal[j].state = ProposalFailed$
$\land\ proposal[j].phase = ProposalApply \Rightarrow proposal[j].state \in \{ProposalComplete,\ ProposalFailed\}$

$IsConsistent(indexes,\ values) \triangleq$
LET
Compute the set of paths in the target that have been updated by transactions
$appliedPaths \triangleq \text{UNION } \{\text{DOMAIN } proposal[i].change.values : i \in indexes\}$
Compute the highest index applied to the target for each path
$pathIndexes \triangleq [p \in appliedPaths \mapsto \text{CHOOSE } i \in indexes :$
$\forall\, j \in indexes :$
$\land\ i \geq j$
$\land\ p \in \text{DOMAIN } proposal[i].change.values]$
Compute the expected target configuration based on the last indexes applied
to the target for each path.
$expectedConfig \triangleq [p \in \text{DOMAIN } pathIndexes \mapsto proposal[pathIndexes[p]].change.values[p]]$
Compute the actual configuration by converting missing path values to *Nil*
$actualConfig \triangleq [p \in \text{DOMAIN } expectedConfig \mapsto \text{IF } p \in \text{DOMAIN } values \text{ THEN } values[p] \text{ ELSE } [value \mapsto$
IN
$actualConfig \neq expectedConfig \Rightarrow \neg(PrintT(indexes) \land PrintT(appliedPaths) \land PrintT(pathIndexes) \land P$

$Consistency \triangleq$
$\land$ LET $indexes \triangleq \{i \in \text{DOMAIN } proposal : \land \lor \land proposal[i].type = ProposalChange$
$\land\ proposal[i].phase = ProposalCommit$
$\land\ proposal[i].state\ = ProposalComplete$
$\lor\ proposal[i].phase = ProposalApply$
$\land \neg\exists\, j \in \text{DOMAIN } proposal :$
$\land\ j > i$
$\land\ proposal[j].type = ProposalRollback$
$\land\ proposal[j].rollback.index = i$
$\land\ proposal[j].phase = ProposalCommit$
$\land\ proposal[j].state\ = ProposalComplete\}$
IN $IsConsistent(indexes,\ configuration.committed.values)$
$\land$ LET $indexes \triangleq \{i \in \text{DOMAIN } proposal : \land proposal[i].type = ProposalChange$
$\land\ proposal[i].phase = ProposalApply$
$\land\ proposal[i].state\ = ProposalComplete$
$\land \neg\exists\, j \in \text{DOMAIN } proposal :$
$\land\ j > i$
$\land\ proposal[j].type = ProposalRollback$
$\land\ proposal[j].rollback.index = i$
$\land\ proposal[j].phase = ProposalApply$
$\land\ proposal[j].state\ = ProposalComplete\}$
IN $IsConsistent(indexes,\ configuration.applied.values)$

$Safety \triangleq \Box(Order \land Consistency)$

THEOREM $Spec \Rightarrow Safety$

$$
\begin{array}{l}
Terminated(i) \triangleq \\
\quad \land\ i \in \text{DOMAIN } proposal \\
\quad \land\ \lor\ \land\ proposal[i].phase = ProposalApply \\
\qquad\quad \land\ proposal[i].state\ = ProposalComplete \\
\qquad \lor\ proposal[i].state = ProposalFailed
\end{array}
$$

$$
\begin{array}{l}
Termination \triangleq \\
\quad \forall\, i \in 1\,..\,Len(proposal) : \\
\qquad Terminated(i)
\end{array}
$$

$$Liveness \triangleq \Diamond Termination$$

THEOREM $Spec \Rightarrow Liveness$

* Modification History
* Last modified *Fri Apr* 21 18:30:03 *PDT* 2023 by *jhalterm*
* Last modified *Mon Feb* 21 01:32:07 *PST* 2022 by *jordanhalterman*
* Created *Wed Sep* 22 13:22:32 *PDT* 2021 by *jordanhalterman*