MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

GenerateTestCases $\triangleq$ FALSE

Nil $\triangleq$ "<nil>"

Change $\triangleq$ "Change"
Rollback $\triangleq$ "Rollback"

Commit $\triangleq$ "Commit"
Apply $\triangleq$ "Apply"

Pending $\triangleq$ "Pending"
InProgress $\triangleq$ "InProgress"
Complete $\triangleq$ "Complete"
Aborted $\triangleq$ "Aborted"
Failed $\triangleq$ "Failed"

Node $\triangleq$ {"node1"}

NumTransactions $\triangleq$ 3

Path $\triangleq$ {"path1"}
Value $\triangleq$ {"value1", "value2"}

A transaction log. Transactions may either request a set
of changes to a set of targets or rollback a prior change.
VARIABLE *transaction*

A record of per-target proposals
VARIABLE *proposal*

A record of per-target configurations
VARIABLE *configuration*

A record of target states
VARIABLE *target*

A record of target masterships

VARIABLE *mastership*

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

LOCAL $Transaction \triangleq$ INSTANCE *Transaction*

LOCAL $Proposal \triangleq$ INSTANCE *Proposal*

LOCAL $Configuration \triangleq$ INSTANCE *Configuration*

LOCAL $Mastership \triangleq$ INSTANCE *Mastership*

$RequestChange(p,\ v) \triangleq$
$\quad \land\ Transaction!RequestChange(p,\ v)$

$RequestRollback(i) \triangleq$
$\quad \land\ Transaction!RequestRollback(i)$

$SetMaster(n) \triangleq$
$\quad \land\ Mastership!SetMaster(n)$
$\quad \land$ UNCHANGED $\langle transaction,\ proposal,\ configuration,\ target \rangle$

$UnsetMaster \triangleq$
$\quad \land\ Mastership!UnsetMaster$
$\quad \land$ UNCHANGED $\langle transaction,\ proposal,\ configuration,\ target \rangle$

$ReconcileTransaction(n,\ i) \triangleq$
$\quad \land\ i \in$ DOMAIN $transaction$
$\quad \land\ Transaction!ReconcileTransaction(n,\ i)$
$\quad \land\ GenerateTestCases \Rightarrow Transaction!Test!Log([node \mapsto n,\ index \mapsto i])$

$ReconcileProposal(n,\ i) \triangleq$
$\quad \land\ i \in$ DOMAIN $proposal$
$\quad \land\ Proposal!ReconcileProposal(n,\ i)$
$\quad \land$ UNCHANGED $\langle transaction \rangle$
$\quad \land\ GenerateTestCases \Rightarrow Proposal!Test!Log([node \mapsto n,\ index \mapsto i])$

$ReconcileConfiguration(n) \triangleq$
$\quad \land\ Configuration!ReconcileConfiguration(n)$
$\quad \land$ UNCHANGED $\langle transaction,\ proposal \rangle$
$\quad \land\ GenerateTestCases \Rightarrow Configuration!Test!Log([node \mapsto n])$

Formal specification, constraints, and theorems.

$$
\begin{array}{l}
Init \triangleq \\
\quad \wedge\ transaction = [\\
\qquad i \in \{\} \mapsto [\\
\qquad\quad type \quad\ \mapsto Change, \\
\qquad\quad index \quad \mapsto 0, \\
\qquad\quad values \ \mapsto [p \in \{\} \mapsto Nil], \\
\qquad\quad commit \mapsto Pending, \\
\qquad\quad apply \quad \mapsto Pending]] \\
\quad \wedge\ proposal = [\\
\qquad i \in \{\} \mapsto [\\
\qquad\quad change \mapsto [\\
\qquad\qquad phase \ \mapsto Nil, \\
\qquad\qquad state \ \mapsto Nil, \\
\qquad\qquad values \mapsto [\\
\qquad\qquad\quad p \in \{\} \ \mapsto [\\
\qquad\qquad\qquad index \mapsto 0, \\
\qquad\qquad\qquad value \mapsto Nil]]], \\
\qquad\quad rollback \mapsto [\\
\qquad\qquad phase \ \mapsto Nil, \\
\qquad\qquad state \ \mapsto Nil, \\
\qquad\qquad values \mapsto [\\
\qquad\qquad\quad p \ \in \{\} \ \mapsto [\\
\qquad\qquad\qquad\quad index \mapsto 0, \\
\qquad\qquad\qquad\quad value \mapsto Nil]]]]] \\
\quad \wedge\ configuration = [\\
\qquad state \ \mapsto InProgress, \\
\qquad term \ \mapsto 0, \\
\qquad committed \mapsto [\\
\qquad\quad index \quad \mapsto 0, \\
\qquad\quad revision \mapsto 0, \\
\qquad\quad values \quad \mapsto [\\
\qquad\qquad p \in \{\} \ \mapsto [\\
\qquad\qquad\quad index \mapsto 0, \\
\qquad\qquad\quad value \mapsto Nil]]], \\
\qquad applied \mapsto [\\
\qquad\quad index \quad \mapsto 0, \\
\qquad\quad revision \mapsto 0, \\
\qquad\quad values \quad \mapsto [\\
\qquad\qquad p \in \{\} \ \mapsto [\\
\qquad\qquad\quad index \mapsto 0, \\
\qquad\qquad\quad value \mapsto Nil]]]] \\
\quad \wedge\ target = [\\
\qquad values \mapsto [\\
\qquad\quad p \in \{\} \ \mapsto [\\
\qquad\quad\ index \mapsto 0,
\end{array}
$$

$$
\begin{array}{l}
\qquad\qquad value \mapsto Nil]]] \\
\quad \land\ mastership = [\\
\qquad\ master \mapsto Nil, \\
\qquad\ term \quad \mapsto 0] \\
\\
Next \triangleq \\
\quad \lor \exists\, p \in Path,\ v \in Value : \\
\qquad RequestChange(p,\ v) \\
\quad \lor \exists\, i \in \text{DOMAIN } transaction : \\
\qquad RequestRollback(i) \\
\quad \lor \exists\, n \in Node : \\
\qquad SetMaster(n) \\
\quad\ \lor \exists\, t \in \text{DOMAIN Target} : \\
\qquad UnsetMaster(t) \\
\quad \lor \exists\, n \in Node : \\
\qquad \exists\, i \in \text{DOMAIN } transaction : \\
\qquad\ ReconcileTransaction(n,\ i) \\
\quad \lor \exists\, n \in Node : \\
\qquad \exists\, i \in \text{DOMAIN } proposal : \\
\qquad\ ReconcileProposal(n,\ i) \\
\quad \lor \exists\, n \in Node : \\
\qquad ReconcileConfiguration(n) \\
\\
Spec \triangleq \\
\quad \land\ Init \\
\quad \land\ \Box[Next]_{vars} \\
\quad \land \forall\, p \in Path,\ v \in Value : \\
\qquad \text{WF}_{\langle transaction,\ proposal,\ configuration,\ mastership,\ target \rangle}(Transaction!RequestChange(p,\ v)) \\
\quad \land \forall\, i \in 1\,..\,NumTransactions : i \in \text{DOMAIN } transaction \Rightarrow \\
\qquad \text{WF}_{\langle transaction,\ proposal,\ configuration,\ mastership,\ target \rangle}(Transaction!RequestRollback(i)) \\
\quad \land \forall\, n \in Node : \\
\qquad \text{WF}_{\langle mastership \rangle}(Mastership!SetMaster(n)) \\
\quad\ \land \exists\, t \in \text{DOMAIN Target} : \\
\qquad \text{WF\_}\langle mastership \rangle(Mastership!UnsetMaster(t)) \\
\quad \land \forall\, n \in Node,\ i \in 1\,..\,NumTransactions : \\
\qquad \text{WF}_{\langle transaction,\ proposal,\ configuration,\ mastership,\ target \rangle}(Transaction!ReconcileTransaction(n,\ i)) \\
\quad \land \forall\, n \in Node,\ i \in 1\,..\,NumTransactions : \\
\qquad \text{WF}_{\langle proposal,\ configuration,\ mastership,\ target \rangle}(Proposal!ReconcileProposal(n,\ i)) \\
\quad \land \forall\, n \in Node : \\
\qquad \text{WF}_{\langle configuration,\ mastership,\ target \rangle}(Configuration!ReconcileConfiguration(n))
\end{array}
$$

$$LimitTransactions \triangleq Len(transaction) \leq NumTransactions$$

$TypeOK \triangleq$
$\quad \land Transaction\,!\,TypeOK$
$\quad \land Proposal\,!\,TypeOK$
$\quad \land Configuration\,!\,TypeOK$
$\quad \land Mastership\,!\,TypeOK$

$Order \triangleq$
$\quad \forall\, i \in \text{DOMAIN } proposal :$
$\quad\quad \land \land proposal[i].phase = Commit$
$\quad\quad\quad \land proposal[i].state \;= InProgress$
$\quad\quad\quad \Rightarrow \neg\exists\, j \in \text{DOMAIN } proposal :$
$\quad\quad\quad\quad\quad \land j > i$
$\quad\quad\quad\quad\quad \land proposal[j].phase = Commit$
$\quad\quad\quad\quad\quad \land proposal[j].state \;= Complete$
$\quad\quad \land \land proposal[i].phase = Apply$
$\quad\quad\quad \land proposal[i].state \;= InProgress$
$\quad\quad\quad \Rightarrow \neg\exists\, j \in \text{DOMAIN } proposal :$
$\quad\quad\quad\quad\quad \land j > i$
$\quad\quad\quad\quad\quad \land proposal[j].phase = Apply$
$\quad\quad\quad\quad\quad \land proposal[j].state \;= Complete$

$Consistency \triangleq$
$\quad \text{LET}$
$\quad\quad$ Compute the transaction indexes that have been applied to the target
$\quad\quad targetIndexes \triangleq \{i \in \text{DOMAIN } transaction :$
$\quad\quad\quad\quad\quad\quad \land i \in \text{DOMAIN } proposal$
$\quad\quad\quad\quad\quad\quad \land proposal[i].phase = Apply$
$\quad\quad\quad\quad\quad\quad \land proposal[i].state \;= Complete$
$\quad\quad\quad\quad\quad\quad \land \neg\exists\, j \in \text{DOMAIN } transaction :$
$\quad\quad\quad\quad\quad\quad\quad\quad \land j > i$
$\quad\quad\quad\quad\quad\quad\quad\quad \land transaction[j].type = Rollback$
$\quad\quad\quad\quad\quad\quad\quad\quad \land transaction[j].rollback = i$
$\quad\quad\quad\quad\quad\quad\quad\quad \land transaction[j].phase = Apply$
$\quad\quad\quad\quad\quad\quad\quad\quad \land transaction[j].state \;= Complete\}$
$\quad\quad$ Compute the set of paths in the target that have been updated by transactions
$\quad\quad appliedPaths \triangleq \text{UNION } \{\text{DOMAIN } proposal[i].change.values : i \in targetIndexes\}$
$\quad\quad$ Compute the highest index applied to the target for each path
$\quad\quad pathIndexes \triangleq [p \in appliedPaths \mapsto \text{CHOOSE } i \in targetIndexes :$
$\quad\quad\quad\quad\quad\quad \forall\, j \in targetIndexes :$
$\quad\quad\quad\quad\quad\quad\quad \land i \geq j$
$\quad\quad\quad\quad\quad\quad\quad \land p \in \text{DOMAIN } proposal[i].change.values]$
$\quad\quad$ Compute the expected target configuration based on the last indexes applied
$\quad\quad$ to the target for each path.
$\quad\quad expectedConfig \triangleq [p \in \text{DOMAIN } pathIndexes \mapsto proposal[pathIndexes[p]].change.values[p]]$
$\quad \text{IN}$

$target = expectedConfig$

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

THEOREM $Spec \Rightarrow Safety$

$Terminated(i) \triangleq$
$\quad \land\ i \in$ DOMAIN $transaction$
$\quad \land\ \lor\ \land\ transaction[i].phase = Apply$
$\qquad\quad\ \land\ transaction[i].state\ = Complete$
$\qquad \lor\ transaction[i].state = Failed$

$Termination \triangleq$
$\quad \forall\, i \in 1\,..\,NumTransactions : \Diamond Terminated(i)$

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$