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"
Done $\triangleq$ {*Complete*, *Aborted*, *Failed*}

Node $\triangleq$ {"node1"}

NumTransactions $\triangleq$ 5

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*

A sequence of state changes used for model checking.
VARIABLE *history*

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

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 $\langle transaction, proposal, configuration, target, history \rangle$

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

$ReconcileTransaction(n, i) \triangleq$
$\quad \land i \in$ 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 $\langle transaction, proposal, history \rangle$
$\quad \land GenerateTestCases \Rightarrow Configuration!Test!Log([node \mapsto n])$

Formal specification, constraints, and theorems.

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

$$
\begin{array}{l}
\quad\quad values \mapsto [\\
\quad\quad\quad p \in \{\} \;\; \mapsto [\\
\quad\quad\quad\quad index \mapsto 0, \\
\quad\quad\quad\quad value \;\mapsto Nil]]] \\
\quad \land\, mastership = [\\
\quad\quad master \mapsto Nil, \\
\quad\quad term \quad\; \mapsto 0] \\
\quad \land\, history = \langle\rangle
\end{array}
$$

$$
\begin{array}{l}
Next \;\triangleq \\
\quad \lor \exists\, p \in Path,\, v \in Value : \\
\quad\quad RequestChange(p,\, v) \\
\quad \lor \exists\, i \in \text{DOMAIN } transaction : \\
\quad\quad RequestRollback(i) \\
\quad \lor \exists\, n \in Node : \\
\quad\quad SetMaster(n) \\
\quad\;\; \lor \exists\, t \in \text{DOMAIN Target} : \\
\quad\quad UnsetMaster(t) \\
\quad \lor \exists\, n \in Node : \\
\quad\quad \exists\, i \in \text{DOMAIN } transaction : \\
\quad\quad\;\; ReconcileTransaction(n,\, i) \\
\quad \lor \exists\, n \in Node : \\
\quad\quad \exists\, i \in \text{DOMAIN } proposal : \\
\quad\quad\;\; ReconcileProposal(n,\, i) \\
\quad \lor \exists\, n \in Node : \\
\quad\quad ReconcileConfiguration(n)
\end{array}
$$

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

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

$TypeOK \triangleq$
- $\land\ Transaction!TypeOK$
- $\land\ Proposal!TypeOK$
- $\land\ Configuration!TypeOK$
- $\land\ Mastership!TypeOK$

LOCAL $IsOrderedChange(p,\ i) \triangleq$
- $\land\quad history[i].type = Change$
- $\land\quad history[i].phase = p$
- $\land\quad \neg\exists\, j \in \text{DOMAIN } history :$
  - $\land\ j < i$
  - $\land\ history[j].type = Change$
  - $\land\ history[j].phase = p$
  - $\land\ history[j].index \geq history[i].index$

LOCAL $IsOrderedRollback(p,\ i) \triangleq$
- $\land\quad history[i].type = Rollback$
- $\land\quad history[i].phase = p$
- $\land\quad \neg\exists\, j \in \text{DOMAIN } history :$
  - $\land\ j < i$
  - $\land\ history[j].type = Change$
  - $\land\ history[j].phase = p$
  - $\land\ history[j].index > history[i].index$
  - $\land\ \neg\exists\, k \in \text{DOMAIN } history :$
    - $\land\ k > j$
    - $\land\ k < i$
    - $\land\ history[k].type = Rollback$
    - $\land\ history[k].phase = p$
    - $\land\ history[k].index = history[j].index$

$Order \triangleq$
- $\land\ \forall\, i \in \text{DOMAIN } history :$
  - $\lor\ IsOrderedChange(Commit,\ i)$
  - $\lor\ IsOrderedChange(Apply,\ i)$
  - $\lor\ IsOrderedRollback(Commit,\ i)$
  - $\lor\ IsOrderedRollback(Apply,\ i)$
- $\land\ \forall\, i \in \text{DOMAIN } proposal :$
  - $\land\ proposal[i].change.phase = Apply$
  - $\land\ proposal[i].change.state\ = Failed$
  - $\land\ proposal[i].rollback.phase = Apply \Rightarrow proposal[i].rollback.state \neq Complete$
  - $\Rightarrow \forall\, j \in \text{DOMAIN } proposal : (j > i \Rightarrow$
    - $(proposal[j].change.phase = Apply \Rightarrow$

$proposal[j].change.state \in \{Nil,\ Pending,\ Aborted\}))$

$Consistency \triangleq$
$\land \forall\, i \in \text{DOMAIN } proposal :$
$\lor\ configuration.committed.index < i$
$\lor\ configuration.committed.revision < i$
$\Rightarrow \neg\exists\, p \in \text{DOMAIN } configuration.committed.values :$
$configuration.committed.values[p].index = i$
$\land \forall\, i \in \text{DOMAIN } proposal :$
$\lor\ configuration.applied.index < i$
$\lor\ configuration.applied.revision < i$
$\Rightarrow\ \land \neg\exists\, p \in \text{DOMAIN } configuration.applied.values :$
$configuration.applied.values[p].index = i$
$\land \neg\exists\, p \in \text{DOMAIN } target.values :$
$target.values[p].index = i$
$\land\ configuration.state = Complete \Rightarrow$
$\forall\, i \in \text{DOMAIN } proposal :$
$\land\ configuration.applied.index \geq i$
$\land\ configuration.applied.revision \geq i$
$\Rightarrow \forall\, p \in \text{DOMAIN } proposal[i].change.values :$
$\land \neg\exists\, j \in \text{DOMAIN } proposal :$
$\land\ j > i$
$\land\ configuration.applied.index \geq j$
$\land\ configuration.applied.revision \geq j$
$\Rightarrow\ \land\ p \in \text{DOMAIN } target.values$
$\land\ target.values[p].value = proposal[i].change.values[p].value$
$\land\ target.values[p].index = proposal[i].change.values[p].index$

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

THEOREM $Spec \Rightarrow Safety$

$Terminates(i) \triangleq$
$\land\ i \in \text{DOMAIN } transaction$
$\land\ transaction[i].commit \in Done$
$\land\ transaction[i].apply \in Done$

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

$Liveness \triangleq Termination$

THEOREM $Spec \Rightarrow Liveness$