MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

GenerateTestCases $\triangleq$ FALSE

Nil $\triangleq$ "<nil>"

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

ReadCommitted $\triangleq$ "ReadCommitted"
Serializable $\triangleq$ "Serializable"

Initialize $\triangleq$ "Initialize"
Validate $\triangleq$ "Validate"
Abort $\triangleq$ "Abort"
Commit $\triangleq$ "Commit"
Apply $\triangleq$ "Apply"

InProgress $\triangleq$ "InProgress"
Complete $\triangleq$ "Complete"
Failed $\triangleq$ "Failed"

Pending $\triangleq$ "Pending"
Validated $\triangleq$ "Validated"
Committed $\triangleq$ "Committed"
Applied $\triangleq$ "Applied"
Aborted $\triangleq$ "Aborted"

Valid $\triangleq$ TRUE
Invalid $\triangleq$ FALSE

Success $\triangleq$ "Success"
Failure $\triangleq$ "Failure"

Node $\triangleq$ {"node-1"}

NumTransactions $\triangleq$ 3

Target $\triangleq$ [
    target1 $\mapsto$ [

```
        $persistent \mapsto$ FALSE,
        $values \mapsto [$
            $path1 \mapsto \{$"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 $Mastership$
```

This section models configuration changes and rollbacks. Changes are appended to the transaction log and processed asynchronously.

```
$Value(s, t, p) \triangleq$
    LET $value \triangleq$ CHOOSE $v \in s : v.target = t \land v.path = p$
    IN
        $[value \mapsto value.value,$
         $delete \mapsto value.delete]$

$Paths(s, t) \triangleq$
    $[p \in \{v.path : v \in \{v \in s : v.target = t\}\} \mapsto Value(s, t, p)]$

$Changes(s) \triangleq$
    $[t \in \{v.target : v \in s\} \mapsto Paths(s, t)]$
```

```
ValidValues(t, p) ≜
   UNION {{[value ↦ v, delete ↦ FALSE] : v ∈ Target[t].values[p]}, {[value ↦ Nil, delete ↦ TRUE]}}

ValidPaths(t) ≜
   UNION {{v @@ [path ↦ p] : v ∈ ValidValues(t, p)} : p ∈ DOMAIN Target[t].values}

ValidTargets ≜
   UNION {{p @@ [target ↦ t] : p ∈ ValidPaths(t)} : t ∈ DOMAIN Target}
```

The set of all valid sets of changes to all targets and their paths.
The set of possible changes is computed from the *Target* model value.

```
ValidChanges ≜
   LET changeSets ≜ {s ∈ SUBSET ValidTargets :
                        ∀ t ∈ DOMAIN Target   :
                          ∀ p ∈ DOMAIN Target[t].values :
                            Cardinality({v ∈ s : v.target = t ∧ v.path = p}) ≤ 1}
   IN
      {Changes(s) : s ∈ changeSets}
```

```
RequestChange(i, c) ≜
   ∧ Transaction!RequestChange(i, c)

RequestRollback(i, j) ≜
   ∧ Transaction!RequestRollback(i, j)

SetMaster(n, t) ≜
   ∧ Mastership!SetMaster(n, t)
   ∧ UNCHANGED ⟨transaction, proposal, configuration, target⟩

UnsetMaster(t) ≜
   ∧ Mastership!UnsetMaster(t)
   ∧ UNCHANGED ⟨transaction, proposal, configuration, target⟩

ReconcileTransaction(n, t) ≜
   ∧ Transaction!ReconcileTransaction(n, t)
   ∧ GenerateTestCases ⇒ Transaction!Test!Log([node ↦ n, index ↦ t])

ReconcileProposal(n, t, i) ≜
   ∧ Proposal!ReconcileProposal(n, t, i)
   ∧ UNCHANGED ⟨transaction⟩
   ∧ GenerateTestCases ⇒ Proposal!Test!Log([node ↦ n, target ↦ t, index ↦ i])

ReconcileConfiguration(n, c) ≜
   ∧ Configuration!ReconcileConfiguration(n, c)
   ∧ UNCHANGED ⟨transaction, proposal⟩
   ∧ GenerateTestCases ⇒ Configuration!Test!Log([node ↦ n, target ↦ c])
```

Formal specification, constraints, and theorems.

```
$Init \triangleq$
    $\land\ transaction = [i \in \{\} \mapsto$
                              $[type \quad \mapsto Change,$
                               $phase \mapsto Initialize,$
                               $state \quad \mapsto InProgress,$
                               $status \mapsto Pending]]$
    $\land\ proposal = [t \in \textsc{domain}\ Target \mapsto$
                          $[i \in \{\} \quad \mapsto$
                            $[phase \mapsto Initialize,$
                             $state \quad \mapsto InProgress]]]$
    $\land\ configuration = [t \in \textsc{domain}\ Target \mapsto$
                               $[state \quad \mapsto InProgress,$
                                $config \mapsto$
                                  $[index \quad \mapsto 0,$
                                   $term \quad \mapsto 0,$
                                   $values \mapsto$
                                      $[path \in \{\} \mapsto$
                                         $[path \qquad \mapsto path,$
                                          $value \quad \mapsto Nil,$
                                          $index \quad \mapsto 0,$
                                          $deleted \mapsto \textsc{false}]]],$
                                $proposal \quad \mapsto [index \mapsto 0],$
                                $commit \quad\ \ \mapsto [index \mapsto 0],$
                                $target \qquad \mapsto$
                                  $[index \quad \mapsto 0,$
                                   $term \quad \mapsto 0,$
                                   $values \mapsto$
                                     $[path \in \{\} \mapsto$
                                        $[path \qquad \mapsto path,$
                                         $value \quad \mapsto Nil,$
                                         $index \quad \mapsto 0,$
                                         $deleted \mapsto \textsc{false}]]]]]$
    $\land\ target = [t \in \textsc{domain}\ Target \mapsto$
                        $[path \in \{\} \mapsto$
                          $[value \mapsto Nil]]]$
    $\land\ mastership = [t \in \textsc{domain}\ Target \mapsto [master \mapsto Nil,\ term \mapsto 0]]$

$Next \triangleq$
    $\lor\ \exists\, i \in 1 \,..\, NumTransactions :$
        $\exists\, c \in ValidChanges :$
          $RequestChange(i,\ c)$
    $\lor\ \exists\, i \in 1 \,..\, NumTransactions :$
```

```
        $\exists j \in \text{DOMAIN } transaction :$
          $RequestRollback(i, j)$
  $\lor \exists n \in Node :$
        $\exists t \in \text{DOMAIN } Target :$
          $SetMaster(n, t)$
  $\lor \exists t \in \text{DOMAIN } Target :$
      $UnsetMaster(t)$
  $\lor \exists n \in Node :$
        $\exists t \in \text{DOMAIN } transaction :$
          $ReconcileTransaction(n, t)$
  $\lor \exists n \in Node :$
        $\exists t \in \text{DOMAIN } proposal :$
          $\exists i \in \text{DOMAIN } proposal[t] :$
            $ReconcileProposal(n, t, i)$
  $\lor \exists n \in Node :$
        $\exists c \in \text{DOMAIN } configuration :$
          $ReconcileConfiguration(n, c)$

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

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

$Consistency \triangleq$
  $\forall t \in \text{DOMAIN } target :$
    LET
        Compute the transaction indexes that have been applied to the target
      $targetIndexes \triangleq \{i \in \text{DOMAIN } transaction :$
                          $\land i \in \text{DOMAIN } proposal[t]$
                          $\land proposal[t][i].phase = Apply$
                          $\land proposal[t][i].state = Complete$
                          $\land t \in transaction[i].targets$
                          $\land \neg\exists j \in \text{DOMAIN } transaction :$
```

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

$Isolation \triangleq$
  $\forall i \in \text{DOMAIN } transaction :$
    $\land \land transaction[i].phase = Commit$
      $\land transaction[i].state = InProgress$
      $\land transaction[i].isolation = Serializable$
      $\Rightarrow \neg \exists j \in \text{DOMAIN } transaction :$
           $\land j > i$
           $\land transaction[j].targets \cap transaction[i].targets \neq \{\}$
           $\land transaction[j].phase = Commit$
    $\land \land transaction[i].phase = Apply$
      $\land transaction[i].state = InProgress$
      $\land transaction[i].isolation = Serializable$
      $\Rightarrow \neg \exists j \in \text{DOMAIN } transaction :$
           $\land j > i$
           $\land transaction[j].targets \cap transaction[i].targets \neq \{\}$
           $\land transaction[j].phase = Apply$

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

$\text{THEOREM } Spec \Rightarrow Safety$

$Terminated(i) \triangleq$
  $\land i \in \text{DOMAIN } transaction$
  $\land transaction[i].phase \in \{Apply, Abort\}$
  $\land transaction[i].state = Complete$

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

$Liveness \triangleq \Diamond Termination$

THEOREM $Spec \Rightarrow Liveness$

* Modification History
* Last modified *Thu Feb* 10 15:59:15 *PST* 2022 by *jordanhalterman*
* Created *Wed Sep* 22 13:22:32 *PDT* 2021 by *jordanhalterman*