
MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

INSTANCE *TLC*

An empty constant

CONSTANT *Nil*

Transaction status constants

CONSTANTS

TransactionPending,
TransactionValidating,
TransactionApplying,
TransactionComplete,
TransactionFailed

Configuration status constants

CONSTANTS

ConfigurationPending,
ConfigurationInitializing,
ConfigurationUpdating,
ConfigurationComplete,
ConfigurationFailed

The set of all nodes

CONSTANT *Node*

Target is the possible targets, paths, and values

Example: $Target \triangleq$ [
 $target1 \mapsto$ [
 $path1 \mapsto \{“value1”, “value2”\}$,
 $path2 \mapsto \{“value2”, “value3”\}$],
 $target2 \mapsto$ [
 $path2 \mapsto \{“value3”, “value4”\}$,
 $path3 \mapsto \{“value4”, “value5”\}$]]

CONSTANT *Target*

ASSUME *Nil* ∈ STRING

ASSUME *TransactionPending* ∈ STRING

ASSUME *TransactionValidating* ∈ STRING

ASSUME *TransactionApplying* ∈ STRING

ASSUME $TransactionComplete \in \text{STRING}$
 ASSUME $TransactionFailed \in \text{STRING}$

 ASSUME $ConfigurationPending \in \text{STRING}$
 ASSUME $ConfigurationInitializing \in \text{STRING}$
 ASSUME $ConfigurationUpdating \in \text{STRING}$
 ASSUME $ConfigurationComplete \in \text{STRING}$
 ASSUME $ConfigurationFailed \in \text{STRING}$

 ASSUME $\wedge IsFiniteSet(Node)$
 $\wedge \forall n \in Node :$
 $\wedge n \notin \text{DOMAIN } Target$
 $\wedge n \in \text{STRING}$

 ASSUME $\wedge \forall t \in \text{DOMAIN } Target :$
 $\wedge t \notin Node$
 $\wedge t \in \text{STRING}$
 $\wedge \forall p \in \text{DOMAIN } Target[t] :$
 $IsFiniteSet(Target[t][p])$

```

TYPE TransactionStatus ::= status ∈
  { TransactionPending,
    TransactionValidating,
    TransactionApplying,
    TransactionComplete,
    TransactionFailed }

TYPE Transaction ≜ [
  id      ::= id ∈ STRING,
  index   ::= index ∈ Nat,
  revision ::= revision ∈ Nat,
  atomic  ::= atomic ∈ BOOLEAN ,
  sync    ::= sync ∈ BOOLEAN ,
  changes ::= [ target ∈ SUBSET (DOMAIN Target) ↦ [
    path ∈ SUBSET (DOMAIN Target[target]) ↦ [
      value ::= value ∈ STRING,
      delete ::= delete ∈ BOOLEAN ]],
  status ::= status ∈ TransactionStatus ]

TYPE ConfigurationStatus ::= status ∈
  { ConfigurationPending,
    ConfigurationInitializing,
    ConfigurationUpdating,
    ConfigurationComplete,
    ConfigurationFailed }

TYPE Configuration ≜ [
  id      ::= id ∈ STRING,
  revision ::= revision ∈ Nat,

```

```

target ::= target ∈ STRING,
paths ::= [ path ∈ SUBSET (DOMAIN Target[target]) ↦ [
    value ::= value ∈ STRING,
    index ::= index ∈ Nat,
    deleted ::= delete ∈ BOOLEAN ]],
txIndex ::= txIndex ∈ Nat,
syncIndex ::= syncIndex ∈ Nat,
mastershipTerm ::= mastershipTerm ∈ Nat,
status ::= status ∈ ConfigurationStatus]

```

A sequence of transactions

Each transactions contains a record of 'changes' for a set of targets

VARIABLE *transactions*

A record of target configurations

Each configuration represents the desired state of the target

VARIABLE *configurations*

A record of target states

VARIABLE *targets*

A record of target masters

VARIABLE *masters*

vars \triangleq $\langle transactions, configurations, targets \rangle$

$ChangeMaster(n, t) \triangleq$
 $\wedge masters[t].master \neq n$
 $\wedge masters' = [masters \text{ EXCEPT } ![t].term = masters[t].term + 1,$
 $\phantom{\wedge masters' = [masters \text{ EXCEPT } } ![t].master = n]$
 $\wedge \text{UNCHANGED } \langle transactions, configurations \rangle$

This section models the northbound *API* for the configuration service.

This crazy thing returns the set of all possible sets of valid changes

$ValidChanges \triangleq$
 LET $allPaths \triangleq \text{UNION } \{ (DOMAIN \ Target[t]) : t \in DOMAIN \ Target \}$
 $allValues \triangleq \text{UNION } \{ \text{UNION } \{ Target[t][p] : p \in DOMAIN \ Target[t] \} : t \in DOMAIN \ Target \}$
 IN
 $\{ targetPathValues \in SUBSET (Target \times allPaths \times allValues \times BOOLEAN) :$
 $\wedge \forall target \in DOMAIN \ Target :$
 LET $targetIndexes \triangleq \{ i \in 1 \dots Len(targetPathValues) : \wedge targetPathValues[i][1] = target \}$
 IN $\vee Cardinality(targetIndexes) = 0$
 $\vee \wedge Cardinality(targetIndexes) = 1$
 $\wedge \text{LET } targetPathValue \triangleq targetPathValues[\text{CHOOSE } index \in targetIndexes : TRUE]$
 $\phantom{\wedge \text{LET } } targetPath \triangleq targetPathValue[2]$

$$\begin{aligned}
& \text{targetValue} \triangleq \text{targetPathValue}[3] \\
& \text{IN} \\
& \wedge \text{targetPath} \setminus (\text{DOMAIN Target}[\text{target}]) = \{\} \\
& \wedge \text{targetValue} \in \text{Target}[\text{target}][\text{targetPath}]
\end{aligned}$$

Add a set of changes to the transaction log

$$\begin{aligned}
\text{Change} & \triangleq \\
& \wedge \exists \text{changes} \in \text{ValidChanges} : \\
& \quad \wedge \text{transactions}' = \text{Append}(\text{transactions}, [\text{index} \mapsto \text{Len}(\text{transactions}) + 1, \\
& \quad \quad \quad \text{atomic} \mapsto \text{FALSE}, \\
& \quad \quad \quad \text{sync} \mapsto \text{FALSE}, \\
& \quad \quad \quad \text{changes} \mapsto \text{changes}, \\
& \quad \quad \quad \text{status} \mapsto \text{TransactionPending}]) \\
& \wedge \text{UNCHANGED} \langle \text{configurations}, \text{targets} \rangle
\end{aligned}$$

This section models the Transaction log reconciler.

Reconcile the transaction log

$$\begin{aligned}
\text{ReconcileTransaction}(n, i) & \triangleq \\
& \text{If the transaction is } \textit{Pending}, \text{ begin validation if the prior transaction} \\
& \text{has already been applied. This simplifies concurrency control in the controller} \\
& \text{and guarantees transactions are applied to the configurations in sequential order.} \\
& \wedge \vee \wedge \text{transactions}[i].\text{status} = \textit{TransactionPending} \\
& \quad \wedge \vee \wedge \text{transactions}[i].\text{index} > 1 \\
& \quad \quad \wedge \text{transactions}[\text{transactions}[i].\text{index} - 1].\text{status} \in \{\textit{TransactionComplete}, \textit{TransactionFailed}\} \\
& \quad \vee \text{transactions}[i].\text{index} = 1 \\
& \quad \wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{transactions}[i].\text{index}].\text{status} = \textit{TransactionValidating}] \\
& \quad \wedge \text{UNCHANGED} \langle \text{configurations} \rangle \\
& \text{If the transaction is in the } \textit{Validating} \text{ state, compute and validate the} \\
& \text{Configuration for each target.} \\
& \vee \wedge \text{transactions}[i].\text{status} = \textit{TransactionValidating} \\
& \quad \text{If validation fails any target, mark the transaction } \textit{Failed}. \\
& \quad \text{If validation is successful, proceed to } \textit{Applying}. \\
& \quad \wedge \exists \text{valid} \in \text{BOOLEAN} : \\
& \quad \quad \vee \wedge \text{valid} \\
& \quad \quad \quad \wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{transactions}[i].\text{index}].\text{status} = \textit{TransactionApplying}] \\
& \quad \quad \vee \wedge \neg \text{valid} \\
& \quad \quad \quad \wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{transactions}[i].\text{index}].\text{status} = \textit{TransactionFailed}] \\
& \quad \wedge \text{UNCHANGED} \langle \text{configurations} \rangle \\
& \text{If the transaction is in the } \textit{Applying} \text{ state, update the Configuration for each} \\
& \text{target and } \textit{Complete} \text{ the transaction.} \\
& \vee \wedge \text{transactions}[i].\text{status} = \textit{TransactionApplying} \\
& \quad \wedge \vee \wedge \text{transactions}[i].\text{atomic} \\
& \quad \quad \text{TODO: Apply atomic transactions here} \\
& \quad \quad \wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{transactions}[i].\text{index}].\text{status} = \textit{TransactionComplete}]
\end{aligned}$$

$$\begin{aligned}
& \wedge \text{UNCHANGED } \langle \text{configurations} \rangle \\
& \vee \wedge \neg \text{transactions}[i].\text{atomic} \\
& \quad \text{Add the transaction index to each updated path} \\
& \wedge \text{configurations}' = [\\
& \quad t \in \text{DOMAIN } \text{Target} \mapsto [\\
& \quad \quad \text{configurations}[t] \text{ EXCEPT} \\
& \quad \quad \quad !.\text{paths} = [\text{path} \in \text{DOMAIN } \text{transactions}[i].\text{changes} \mapsto \\
& \quad \quad \quad \quad \text{transactions}[i].\text{changes}[\text{path}] @@ [\text{index} \mapsto \text{transactions}[i].\text{index}] @@ \text{configurations}[t] \\
& \quad \quad \quad !.\text{txIndex} = \text{transactions}[i].\text{index}, \\
& \quad \quad \quad !.\text{status} = \text{ConfigurationPending}] \\
& \quad \wedge \text{transactions}' = [\text{transactions} \text{ EXCEPT } ![\text{transactions}[i].\text{index}].\text{status} = \text{TransactionComplete}] \\
& \wedge \text{UNCHANGED } \langle \text{targets} \rangle
\end{aligned}$$

This section models the Configuration reconciler.

$$\begin{aligned}
& \text{ReconcileConfiguration}(n, c) \triangleq \\
& \wedge \vee \wedge \text{configurations}[c].\text{status} = \text{ConfigurationPending} \\
& \quad \text{If the configuration is marked } \text{ConfigurationPending} \text{ and mastership} \\
& \quad \text{has changed (indicated by an increased mastership term), mark the} \\
& \quad \text{configuration } \text{ConfigurationInitializing} \text{ to force full re-synchronization.} \\
& \wedge \vee \wedge \text{masters}[\text{configurations}[c].\text{target}].\text{term} > \text{configurations}[c].\text{mastershipTerm} \\
& \quad \wedge \text{configurations}' = [\text{configurations} \text{ EXCEPT } ![\text{c}].\text{status} = \text{ConfigurationInitializing}, \\
& \quad \quad \quad ![\text{c}].\text{mastershipTerm} = \text{masters}[\text{configurations}[c].\text{target}].\text{term}] \\
& \quad \text{If the configuration is marked } \text{ConfigurationPending} \text{ and the values have} \\
& \quad \text{changed (determined by comparing the transaction index to the last } \text{sync} \\
& \quad \text{index), mark the configuration } \text{ConfigurationUpdating} \text{ to push the changes} \\
& \quad \text{to the target.} \\
& \vee \wedge \text{configurations}[c].\text{syncIndex} < \text{configurations}[c].\text{txIndex} \\
& \quad \wedge \text{configurations}' = [\text{configurations} \text{ EXCEPT } ![\text{c}].\text{status} = \text{ConfigurationUpdating}] \\
& \vee \wedge \text{configurations}[c].\text{status} = \text{ConfigurationInitializing} \\
& \quad \wedge \text{masters}[\text{configurations}[c].\text{target}].\text{master} = n \\
& \quad \text{Merge the configuration paths with the target paths, removing paths} \\
& \quad \text{that have been marked deleted} \\
& \wedge \text{LET } \text{deletePaths} \triangleq \{p \in \text{DOMAIN } \text{configurations}[c].\text{paths} : \text{configurations}[c].\text{paths}[p].\text{deleted}\} \\
& \quad \text{IN} \\
& \quad \wedge \text{targets}' = [\text{targets} \text{ EXCEPT } ![\text{configurations}[c].\text{target}] = \\
& \quad \quad [p \in (\text{DOMAIN } \text{c.paths} \setminus \text{deletePaths}) \mapsto [\text{value} \mapsto \text{configurations}[c].\text{paths}[p]]] @@ \\
& \quad \quad [p \in (\text{DOMAIN } \text{targets}[\text{configurations}[c].\text{target}] \setminus \text{deletePaths}) \mapsto \text{targets}[\text{configurations}[c].\text{target}]]] \\
& \quad \text{Set the configuration's status to } \text{Complete} \\
& \quad \wedge \text{configurations}' = [\text{configurations} \text{ EXCEPT } ![\text{c}].\text{status} = \text{ConfigurationComplete}, \\
& \quad \quad \quad ![\text{c}].\text{syncIndex} = \text{configurations}[c].\text{txIndex}] \\
& \quad \text{If the configuration is marked } \text{ConfigurationUpdating}, \text{ we only need to} \\
& \quad \text{push paths that have changed since the target was initialized or last} \\
& \quad \text{updated by the controller. The set of changes made since the last}
\end{aligned}$$

synchronization are identified by comparing the index of each path-value
to the last synchronization index, *syncIndex*

$\vee \wedge \text{configurations}[c].\text{status} = \text{ConfigurationUpdating}$
 $\wedge \text{masters}[\text{configurations}[c].\text{target}].\text{master} = n$
*Compute the set of updated and deleted paths by comparing
their indexes to the target's last sync index.*

$\wedge \text{LET } \text{updatedPaths} \triangleq \{p \in \text{DOMAIN } \text{configurations}[c].\text{paths} : \text{configurations}[c].\text{paths}[p].\text{index} > \text{syncIndex}\}$
 $\text{deletedPaths} \triangleq \{p \in \text{updatedPaths} : \text{configurations}[c].\text{paths}[p].\text{deleted}\}$

IN

*Update the target paths by adding/updating paths that have changed and
removing paths that have been deleted since the last sync.*

$\wedge \text{targets}' = [\text{targets} \text{ EXCEPT } ![\text{configurations}[c].\text{target}] =$
 $\quad [p \in \text{updatedPaths} \setminus \text{deletedPaths} \mapsto \text{configurations}[c].\text{paths}[p]] @@$
 $\quad [p \in \text{DOMAIN } \text{targets}[\text{configurations}[c].\text{target}] \setminus \text{deletedPaths} \mapsto \text{targets}[\text{configurations}[c].\text{target}][p]]$

$\wedge \text{configurations}' = [\text{configurations} \text{ EXCEPT } ![\text{c}].\text{status} = \text{ConfigurationComplete},$
 $\quad ![\text{c}].\text{syncIndex} = \text{configurations}[c].\text{txIndex}]$

*If the configuration is not already ConfigurationPending and mastership
has been lost revert it. This can occur when the connection to the
target has been lost and the mastership is no longer valid.*

TODO: We still need to model mastership changes

$\vee \wedge c.\text{status} \neq \text{ConfigurationPending}$
 $\wedge \text{masters}[\text{configurations}[c].\text{target}].\text{master} = \text{Nil}$
 $\wedge \text{configurations}' = [\text{configurations} \text{ EXCEPT } ![\text{c}].\text{status} = \text{ConfigurationPending}]$

$\wedge \text{UNCHANGED } \langle \text{transactions} \rangle$

Init and next state predicates

Init \triangleq

$\wedge \text{transactions} = \langle \rangle$
 $\wedge \text{configurations} = [t \in \text{Target} \mapsto$
 $\quad [id \mapsto t,$
 $\quad \text{config} \mapsto$
 $\quad \quad [path \in \{\} \mapsto$
 $\quad \quad \quad [path \mapsto path,$
 $\quad \quad \quad value \mapsto \text{Nil},$
 $\quad \quad \quad index \mapsto 0,$
 $\quad \quad \quad deleted \mapsto \text{FALSE}]]]]$

$\wedge \text{targets} = [t \in \text{Target} \mapsto$
 $\quad [path \in \{\} \mapsto$
 $\quad \quad [value \mapsto \text{Nil}]]]$

$\wedge \text{masters} = [t \in \text{Target} \mapsto [master \mapsto \text{Nil}, term \mapsto 0]]$

Next \triangleq

$\vee \text{Change}$
 $\vee \exists n \in \text{Node} :$

$$\begin{aligned}
& \forall \exists t \in \text{DOMAIN } \textit{Target} : \\
& \quad \textit{ChangeMaster}(n, t) \\
& \forall \exists t \in \text{DOMAIN } \textit{transactions} : \\
& \quad \textit{ReconcileTransaction}(n, t) \\
& \forall \exists t \in \text{DOMAIN } \textit{configurations} : \\
& \quad \textit{ReconcileConfiguration}(n, t)
\end{aligned}$$

$$\textit{Spec} \triangleq \textit{Init} \wedge \Box[\textit{Next}]_{\textit{vars}}$$

\ * Modification History
\ * Last modified Sun Jan 16 10:00:11 PST 2022 by jordanhalterman
\ * Created Wed Sep 22 13:22:32 PDT 2021 by jordanhalterman