
MODULE *ConfigImpl*

INSTANCE *Naturals*
 INSTANCE *FiniteSets*
 INSTANCE *Sequences*
 LOCAL INSTANCE *TLC*

This section specifies constant parameters for the model.
 CONSTANT *LogEnabled*
 ASSUME *LogEnabled* ∈ BOOLEAN
 CONSTANT *None*
 ASSUME *None* ∈ STRING
 CONSTANT *Node*
 ASSUME $\forall n \in \text{Node} : n \in \text{STRING}$
 CONSTANTS
 Change,
 Rollback
Event $\triangleq \{ \text{Change}, \text{Rollback} \}$
 ASSUME $\forall e \in \text{Event} : e \in \text{STRING}$
 CONSTANTS
 Commit,
 Apply
Phase $\triangleq \{ \text{Commit}, \text{Apply} \}$
 ASSUME $\forall p \in \text{Phase} : p \in \text{STRING}$
 CONSTANTS
 Pending,
 InProgress,
 Complete,
 Aborted,
 Failed
State $\triangleq \{ \text{Pending}, \text{InProgress}, \text{Complete}, \text{Aborted}, \text{Failed} \}$
Done $\triangleq \{ \text{Complete}, \text{Aborted}, \text{Failed} \}$

ASSUME $\forall s \in State : s \in \text{STRING}$

CONSTANT *Path*

ASSUME $\forall p \in Path : p \in \text{STRING}$

CONSTANT *Value*

ASSUME $\forall v \in Value : v \in \text{STRING}$

$AllValues \triangleq Value \cup \{None\}$

CONSTANT *NumProposals*

ASSUME $NumProposals \in Nat$

This section defines model state variables.

$proposal \triangleq [i \in 1 \dots Nat \mapsto [$
 $phase \mapsto Phase,$
 $change \mapsto [$
 $values \mapsto Change,$
 $commit \mapsto State,$
 $apply \mapsto State],$
 $rollback \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change,$
 $commit \mapsto State,$
 $apply \mapsto State]]]$

$configuration \triangleq [$
 $committed \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change],$
 $applied \mapsto [$
 $index \mapsto Nat,$
 $values \mapsto Change,$
 $term \mapsto Nat]]]$

$mastership \triangleq [$
 $master \mapsto \text{STRING},$
 $term \mapsto Nat,$
 $conn \mapsto Nat]$

$conn \triangleq [n \in Node \mapsto [$
 $id \mapsto Nat,$
 $connected \mapsto \text{BOOLEAN}]]$

$target \triangleq [$
 $id \mapsto Nat,$
 $values \mapsto Change,$
 $running \mapsto \text{BOOLEAN}]$

VARIABLE *proposal*

VARIABLE *configuration*

VARIABLE *mastership*

VARIABLE *conn*

VARIABLE *target*

VARIABLE *history*

VARIABLE *mapping*

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

LOCAL *MastershipLog* \triangleq INSTANCE *Log* WITH

File \leftarrow "Mastership.log",

CurrState \leftarrow [

target $\mapsto target$,

mastership $\mapsto mastership$,

conns $\mapsto conn$],

SuccState \leftarrow [

target $\mapsto target'$,

mastership $\mapsto mastership'$,

conns $\mapsto conn'$],

Enabled $\leftarrow LogEnabled$

LOCAL *ConfigurationLog* \triangleq INSTANCE *Log* WITH

File \leftarrow "Configuration.log",

CurrState \leftarrow [

configuration $\mapsto configuration$,

target $\mapsto target$,

mastership $\mapsto mastership$,

conns $\mapsto conn$],

SuccState \leftarrow [

configuration $\mapsto configuration'$,

target $\mapsto target'$,

mastership $\mapsto mastership'$,

conns $\mapsto conn'$],

Enabled $\leftarrow LogEnabled$

LOCAL *ProposalLog* \triangleq INSTANCE *Log* WITH

File \leftarrow "Proposal.log",

CurrState \leftarrow [

proposals $\mapsto [i \in \{i \in \text{DOMAIN } proposal : proposal[i].phase \neq None\} \mapsto proposal[i]]$,

$$\begin{aligned}
& configuration \mapsto configuration, \\
& target \mapsto target, \\
& mastership \mapsto mastership, \\
& conns \mapsto conn], \\
SuccState \leftarrow [& \\
& proposals \mapsto [i \in \{i \in \text{DOMAIN } proposal' : proposal'[i].phase \neq \text{None}\} \mapsto proposal'[i]], \\
& configuration \mapsto configuration', \\
& target \mapsto target', \\
& mastership \mapsto mastership', \\
& conns \mapsto conn'], \\
Enabled \leftarrow & LogEnabled
\end{aligned}$$

This section models configuration target.

$$\begin{aligned}
StartTarget & \triangleq \\
& \wedge \neg target.running \\
& \wedge target' = [target \text{ EXCEPT } !.id = target.id + 1, \\
& \quad !.running = \text{TRUE}] \\
& \wedge \text{UNCHANGED } \langle proposal, configuration, mastership, conn, history \rangle \\
StopTarget & \triangleq \\
& \wedge target.running \\
& \wedge target' = [target \text{ EXCEPT } !.running = \text{FALSE}, \\
& \quad !.values = [p \in \{\} \mapsto [value \mapsto \text{None}]]] \\
& \wedge conn' = [n \in Node \mapsto [conn[n] \text{ EXCEPT } !.connected = \text{FALSE}]] \\
& \wedge \text{UNCHANGED } \langle proposal, configuration, mastership, history \rangle
\end{aligned}$$

This section models nodes connection to the configuration target.

$$\begin{aligned}
ConnectNode(n) & \triangleq \\
& \wedge \neg conn[n].connected \\
& \wedge target.running \\
& \wedge conn' = [conn \text{ EXCEPT } ![n].id = conn[n].id + 1, \\
& \quad ![n].connected = \text{TRUE}] \\
& \wedge \text{UNCHANGED } \langle proposal, configuration, mastership, target, history \rangle \\
DisconnectNode(n) & \triangleq \\
& \wedge conn[n].connected \\
& \wedge conn' = [conn \text{ EXCEPT } ![n].connected = \text{FALSE}] \\
& \wedge \text{UNCHANGED } \langle proposal, configuration, mastership, target, history \rangle
\end{aligned}$$

This section models *mastership* reconciliation.

If the index is less than the *targetIndex*, this indicates a rollback of a prior proposal is being processed, and the *targetIndex* cannot be incremented until that rollback is complete. The index represents the index to which the proposal at *changeIndex* + 1 rolls back.

7

$CommitRollback(n, i) \triangleq$

'index' is the current index committed to the configuration
 'changeIndex' is the maximum change index committed to the configuration
 'targetIndex' is the index of the proposal currently being committed
targetIndex is always changed first. Once the rollback is committed, the
 index will be decremented to match the *targetIndex*. The next time a change
 is committed, the index will increase again. If the committed index is equal
 to this proposal index, this proposal is the next to be rolled back. To roll
 back a proposal, the target index is set to the proposal's rollback index.
 When the rollback is committed, the committed index is set to the proposal's
 rollback index, thus matching the *targetIndex*. This unblocks new changes
 to be committed.

$\wedge \vee \wedge proposal[i].rollback.commit = Pending$
 $\wedge configuration.committed.changeIndex \geq i$
 $\wedge configuration.committed.index = i$
 $\wedge \vee \wedge configuration.committed.targetIndex = i$
 $\wedge configuration' = [configuration \text{ EXCEPT } !.committed.targetIndex = proposal[i].rollback.index]$
 $\wedge UNCHANGED \langle proposal \rangle$
 $\vee \wedge configuration.committed.targetIndex = proposal[i].rollback.index$
 $\wedge \vee \wedge proposal[i].change.commit \neq Aborted$
 $\wedge proposal' = [proposal \text{ EXCEPT } ![i].rollback.commit = InProgress]$
 $\vee \wedge proposal[i].change.commit = Aborted$
 $\wedge proposal' = [proposal \text{ EXCEPT } ![i].rollback.commit = Complete]$
 $\wedge UNCHANGED \langle configuration \rangle$
 $\wedge UNCHANGED \langle history \rangle$
 $\vee \wedge proposal[i].rollback.commit = InProgress$
 $\wedge \vee \wedge configuration.committed.index = i$
 $\wedge configuration' = [configuration \text{ EXCEPT } !.committed.index = proposal[i].rollback.index,$
 $! .committed.values = proposal[i].rollback.values @@$
 $configuration.committed.values]$
 $\wedge history' = Append(history, [type \mapsto Rollback, phase \mapsto Commit, index \mapsto i])$
 $\wedge UNCHANGED \langle proposal \rangle$
 $\vee \wedge configuration.committed.index = proposal[i].rollback.index$
 $\wedge proposal' = [proposal \text{ EXCEPT } ![i].rollback.commit = Complete]$
 $\wedge UNCHANGED \langle configuration, history \rangle$
 $\vee \wedge proposal[i].rollback.commit = Complete$
 $\wedge configuration.committed.targetIndex = proposal[i].rollback.index$
 $\wedge configuration.committed.index \neq proposal[i].rollback.index$
 $\wedge configuration' = [configuration \text{ EXCEPT } !.committed.index = proposal[i].rollback.index]$
 $\wedge UNCHANGED \langle proposal, history \rangle$
 $\wedge UNCHANGED \langle target \rangle$

$ApplyRollback(n, i) \triangleq$

'index' is the current index applied to the configuration
 'changeIndex' is the maximum change index applied to the configuration

'targetIndex' is the index of the proposal currently being applied
targetIndex is always changed first. Once the rollback is applied, the
index will be decremented to match the *targetIndex*. The next time a change
is applied, the index will increase again. If the applied index is equal
to this proposal index, this proposal is the next to be rolled back. To roll
back a proposal, the target index is set to the proposal's rollback index.
When the rollback is applied, the applied index is set to the proposal's
rollback index, thus matching the *targetIndex*. This unblocks new changes
to be applied.

$$\begin{aligned}
& \wedge \vee \wedge \text{proposal}[i].\text{rollback.apply} = \text{Pending} \\
& \quad \wedge \text{configuration.committed.index} \leq \text{proposal}[i].\text{rollback.index} \\
& \quad \wedge \text{configuration.applied.changeIndex} \geq i \\
& \quad \wedge \text{configuration.applied.index} = i \\
& \quad \wedge \vee \wedge \text{configuration.applied.targetIndex} = i \\
& \quad \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.targetIndex} = \text{proposal}[i].\text{rollback.index}] \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \quad \vee \wedge \text{configuration.applied.targetIndex} = \text{proposal}[i].\text{rollback.index} \\
& \quad \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{rollback.apply} = \text{InProgress}] \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{configuration} \rangle \\
& \quad \wedge \text{UNCHANGED } \langle \text{target}, \text{history} \rangle \\
& \vee \wedge \text{proposal}[i].\text{rollback.apply} = \text{InProgress} \\
& \quad \wedge \vee \wedge \text{configuration.applied.index} = i \\
& \quad \quad \text{Verify the applied term is the current } \textit{mastership} \text{ term to ensure the} \\
& \quad \quad \text{configuration has been synchronized following restarts.} \\
& \quad \quad \wedge \text{configuration.applied.term} = \text{mastership.term} \\
& \quad \quad \text{Verify the node's connection to the target.} \\
& \quad \quad \wedge \text{conn}[n].\text{connected} \\
& \quad \quad \wedge \text{target.running} \\
& \quad \quad \wedge \text{target}' = [\text{target} \text{ EXCEPT } !.\text{values} = \text{proposal}[i].\text{rollback.values} @ @ \text{target.values}] \\
& \quad \quad \wedge \text{configuration}' = [\text{configuration} \text{ EXCEPT } !.\text{applied.index} = \text{proposal}[i].\text{rollback.index}, \\
& \quad \quad \quad !.\text{applied.values} = \text{proposal}[i].\text{rollback.values} @ @ \\
& \quad \quad \quad \text{configuration.applied.values}] \\
& \quad \quad \wedge \text{history}' = \text{Append}(\text{history}, [\text{type} \mapsto \text{Rollback}, \text{phase} \mapsto \text{Apply}, \text{index} \mapsto i]) \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{proposal} \rangle \\
& \vee \wedge \text{configuration.applied.index} \neq i \\
& \quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } ![i].\text{rollback.apply} = \text{Complete}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{configuration}, \text{target}, \text{history} \rangle
\end{aligned}$$

$$\begin{aligned}
& \text{ReconcileProposal}(n, i) \triangleq \\
& \quad \wedge \text{mastership.master} = n \\
& \quad \wedge \vee \text{CommitChange}(n, i) \\
& \quad \quad \vee \text{ApplyChange}(n, i) \\
& \quad \quad \vee \text{CommitRollback}(n, i) \\
& \quad \quad \vee \text{ApplyRollback}(n, i) \\
& \quad \wedge \text{UNCHANGED } \langle \text{mastership}, \text{conn} \rangle
\end{aligned}$$

This section models changes to the proposal queue.

Propose change at index 'i'

$$\begin{aligned}
\text{ProposeChange}(i) &\triangleq \\
&\wedge \text{proposal}[i].\text{phase} = \text{None} \\
&\wedge i - 1 \in \text{DOMAIN } \text{proposal} \Rightarrow \text{proposal}[i - 1].\text{phase} \neq \text{None} \\
&\wedge \exists p \in \text{Path}, v \in \text{AllValues} : \\
&\quad \wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } \begin{array}{ll} ! [i].\text{phase} & = \text{Change}, \\ ! [i].\text{change.values} & = (p :> [value \mapsto v]), \\ ! [i].\text{change.commit} & = \text{Pending}, \\ ! [i].\text{change.apply} & = \text{Pending} \end{array} \\
&\quad \wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle
\end{aligned}$$

Rollback proposed change at index 'i'

$$\begin{aligned}
\text{ProposeRollback}(i) &\triangleq \\
&\wedge \text{proposal}[i].\text{phase} = \text{Change} \\
&\wedge \text{proposal}' = [\text{proposal} \text{ EXCEPT } \begin{array}{ll} ! [i].\text{phase} & = \text{Rollback}, \\ ! [i].\text{rollback.commit} & = \text{Pending}, \\ ! [i].\text{rollback.apply} & = \text{Pending} \end{array} \\
&\quad \wedge \text{UNCHANGED } \langle \text{configuration}, \text{mastership}, \text{conn}, \text{target}, \text{history} \rangle
\end{aligned}$$

Formal specification, constraints, and theorems.

$$\begin{aligned}
\text{Init} &\triangleq \\
&\wedge \text{proposal} = [\\
&\quad i \in 1 \dots \text{NumProposals} \mapsto [\\
&\quad \quad \text{phase} \mapsto \text{None}, \\
&\quad \quad \text{change} \mapsto [\\
&\quad \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
&\quad \quad \quad \text{commit} \mapsto \text{None}, \\
&\quad \quad \quad \text{apply} \mapsto \text{None}], \\
&\quad \quad \text{rollback} \mapsto [\\
&\quad \quad \quad \text{index} \mapsto 0, \\
&\quad \quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
&\quad \quad \quad \text{commit} \mapsto \text{None}, \\
&\quad \quad \quad \text{apply} \mapsto \text{None}]] \\
&\wedge \text{configuration} = [\\
&\quad \text{committed} \mapsto [\\
&\quad \quad \text{index} \mapsto 0, \\
&\quad \quad \text{changeIndex} \mapsto 0, \\
&\quad \quad \text{targetIndex} \mapsto 0, \\
&\quad \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]]], \\
&\quad \text{applied} \mapsto [\\
&\quad \quad \text{index} \mapsto 0,
\end{aligned}$$

$$\begin{aligned}
& \text{changeIndex} \mapsto 0, \\
& \text{targetIndex} \mapsto 0, \\
& \text{term} \mapsto 0, \\
& \text{target} \mapsto 0, \\
& \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
& \text{status} \mapsto \text{Pending}] \\
\wedge \text{mastership} &= [\text{master} \mapsto \text{None}, \text{term} \mapsto 0, \text{conn} \mapsto 0] \\
\wedge \text{conn} &= [n \in \text{Node} \mapsto [\text{id} \mapsto 0, \text{connected} \mapsto \text{FALSE}]] \\
\wedge \text{target} &= [\\
& \quad \text{id} \mapsto 0, \\
& \quad \text{values} \mapsto [p \in \{\} \mapsto [\text{index} \mapsto 0, \text{value} \mapsto \text{None}]], \\
& \quad \text{running} \mapsto \text{FALSE}] \\
\wedge \text{history} &= \langle \rangle \\
\wedge \text{mapping} &= [\\
& \quad \text{configuration} \mapsto [\\
& \quad \quad \text{committed} \mapsto [\\
& \quad \quad \quad \text{values} \mapsto \text{configuration.committed.values}, \\
& \quad \quad \text{applied} \mapsto [\\
& \quad \quad \quad \text{term} \mapsto \text{configuration.applied.term}, \\
& \quad \quad \quad \text{target} \mapsto \text{configuration.applied.target}, \\
& \quad \quad \quad \text{values} \mapsto \text{configuration.applied.values}, \\
& \quad \quad \text{status} \mapsto \text{configuration.status}], \\
& \quad \text{proposal} \mapsto [i \in \text{DOMAIN proposal} \mapsto [\\
& \quad \quad \text{phase} \mapsto \text{proposal}[i].\text{phase}, \\
& \quad \quad \text{values} \mapsto [p \in \text{DOMAIN proposal}[i].\text{change.values} \mapsto \text{proposal}[i].\text{change.values}[p].\text{value}], \\
& \quad \quad \text{change} \mapsto [\\
& \quad \quad \quad \text{commit} \mapsto \text{IF } \wedge \text{proposal}[i].\text{change.commit} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration.committed.changeIndex} \geq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE proposal}[i].\text{change.commit}, \\
& \quad \quad \text{apply} \mapsto \text{IF } \wedge \text{proposal}[i].\text{change.apply} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration.applied.changeIndex} \geq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE proposal}[i].\text{change.apply}], \\
& \quad \text{rollback} \mapsto [\\
& \quad \quad \text{commit} \mapsto \text{IF } \wedge \text{proposal}[i].\text{rollback.commit} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration.committed.index} \neq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE proposal}[i].\text{rollback.commit}, \\
& \quad \quad \text{apply} \mapsto \text{IF } \wedge \text{proposal}[i].\text{rollback.commit} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration.applied.index} \neq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE proposal}[i].\text{rollback.apply}]]]]]
\end{aligned}$$

Next \triangleq

$$\begin{aligned}
& \wedge \vee \exists i \in 1 \dots \text{NumProposals} : \\
& \quad \vee \text{ProposeChange}(i) \\
& \quad \vee \text{ProposeRollback}(i) \\
& \vee \exists n \in \text{Node}, i \in \text{DOMAIN } \text{proposal} : \\
& \quad \text{ProposalLog!Action}(\text{ReconcileProposal}(n, i), [\text{node} \mapsto n, \text{index} \mapsto i]) \\
& \vee \exists n \in \text{Node} : \\
& \quad \text{ConfigurationLog!Action}(\text{ReconcileConfiguration}(n), [\text{node} \mapsto n]) \\
& \vee \exists n \in \text{Node} : \\
& \quad \text{MastershipLog!Action}(\text{ReconcileMastership}(n), [\text{node} \mapsto n]) \\
& \vee \exists n \in \text{Node} : \\
& \quad \vee \text{ConnectNode}(n) \\
& \quad \vee \text{DisconnectNode}(n) \\
& \vee \text{StartTarget} \\
& \vee \text{StopTarget} \\
& \wedge \text{mapping}' = [\\
& \quad \text{configuration} \mapsto [\\
& \quad \quad \text{committed} \mapsto [\\
& \quad \quad \quad \text{values} \mapsto \text{configuration}'.\text{committed}.\text{values}], \\
& \quad \quad \text{applied} \mapsto [\\
& \quad \quad \quad \text{term} \mapsto \text{configuration}'.\text{applied}.\text{term}, \\
& \quad \quad \quad \text{target} \mapsto \text{configuration}'.\text{applied}.\text{target}, \\
& \quad \quad \quad \text{values} \mapsto \text{configuration}'.\text{applied}.\text{values}], \\
& \quad \quad \text{status} \mapsto \text{configuration}'.\text{status}], \\
& \quad \text{proposal} \mapsto [i \in \text{DOMAIN } \text{proposal}' \mapsto [\\
& \quad \quad \text{phase} \mapsto \text{proposal}'[i].\text{phase}, \\
& \quad \quad \text{values} \mapsto [p \in \text{DOMAIN } \text{proposal}'[i].\text{change}.\text{values} \mapsto \text{proposal}'[i].\text{change}.\text{values}[p].\text{value}], \\
& \quad \quad \text{change} \mapsto [\\
& \quad \quad \quad \text{commit} \mapsto \text{IF } \wedge \text{proposal}'[i].\text{change}.\text{commit} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration}'.\text{committed}.\text{changeIndex} \geq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE } \text{proposal}'[i].\text{change}.\text{commit}, \\
& \quad \quad \quad \text{apply} \mapsto \text{IF } \wedge \text{proposal}'[i].\text{change}.\text{apply} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration}'.\text{applied}.\text{changeIndex} \geq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE } \text{proposal}'[i].\text{change}.\text{apply}], \\
& \quad \quad \text{rollback} \mapsto [\\
& \quad \quad \quad \text{commit} \mapsto \text{IF } \wedge \text{proposal}'[i].\text{rollback}.\text{commit} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration}'.\text{committed}.\text{index} \neq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE } \text{proposal}'[i].\text{rollback}.\text{commit}, \\
& \quad \quad \quad \text{apply} \mapsto \text{IF } \wedge \text{proposal}'[i].\text{rollback}.\text{apply} = \text{InProgress} \\
& \quad \quad \quad \quad \wedge \text{configuration}'.\text{applied}.\text{index} \neq i \\
& \quad \quad \quad \quad \text{THEN Complete} \\
& \quad \quad \quad \quad \text{ELSE } \text{proposal}'[i].\text{rollback}.\text{apply}]]]]
\end{aligned}$$

$$\begin{aligned}
Spec &\triangleq \\
&\wedge Init \\
&\wedge \Box[Next]_{vars} \\
&\wedge \forall i \in 1 \dots NumProposals : WF_{\langle proposal, configuration, mastership, conn, target, history \rangle} (ProposeChange(i) \vee ProposeChange(i)) \\
&\wedge \forall n \in Node, i \in 1 \dots NumProposals : WF_{\langle proposal, configuration, mastership, conn, target, history \rangle} (ReconcileProposal(i, n)) \\
&\wedge \forall n \in Node : WF_{\langle configuration, mastership, conn, target \rangle} (ReconcileConfiguration(n)) \\
&\wedge \forall n \in Node : WF_{\langle mastership, conn, target \rangle} (ReconcileMastership(n)) \\
&\wedge \forall n \in Node : WF_{\langle conn, target \rangle} (ConnectNode(n) \vee DisconnectNode(n)) \\
&\wedge WF_{\langle target \rangle} (StartTarget) \\
&\wedge WF_{\langle target \rangle} (StopTarget) \\
Mapping &\triangleq \text{INSTANCE } Config \text{ WITH} \\
&\quad proposal \leftarrow mapping.proposal, \\
&\quad configuration \leftarrow mapping.configuration \\
Refinement &\triangleq Mapping!Spec \\
Order &\triangleq Mapping!Order \\
Consistency &\triangleq Mapping!Consistency \\
Liveness &\triangleq Mapping!Liveness
\end{aligned}$$
