```
──────────────── MODULE ViewstampedReplication ────────────────
EXTENDS Integers, Sequences, FiniteSets

CONSTANTS Replica, Quorum

 Replica Status
CONSTANTS Normal, ViewChange, Recovering

 Client operation
CONSTANT Operation

 types of log blocks
CONSTANTS RequestBlock, ViewBlock

 Special value
CONSTANT None

 Message types for processing logs
CONSTANTS DownloadChunk, StartDownload, PrepareOk, Commit

 Message types for view changing
CONSTANTS StartViewChange, DoViewChange, StartView

 Sequence with all replicas (for view selection)
CONSTANT ReplicaSequence

 For state space limitation
CONSTANT MaxRequests, MaxViews

 State on each replica
VARIABLE replicaState

VARIABLE msgs

vars ≜ ⟨replicaState, msgs⟩

───────────────────────────────────────────────────────────────

LogEntry ≜ [type : {RequestBlock}, opNumber : Nat, op : Operation]
         ∪ [type       : {ViewBlock}, view : Nat]

 All possible messages
Message ≜ [type : {DownloadChunk}, v : Nat, m : LogEntry, n : Nat, k : Nat, i : Replica]
        ∪ [type       : {StartDownload}, v   : Nat, n  : Nat, src : Replica]
        ∪ [type       : {PrepareOk}, v : Nat, n : Nat, i : Replica]
        ∪ [type       : {Commit}, v : Nat, k : Nat]
        ∪ [type       : {StartViewChange}, v : Nat, i : Replica]
        ∪ [type       : {DoViewChange}, v : Nat, vv : Nat,
            n : Nat, k : Nat, i : Replica]
        ∪ [type : {StartView}, v : Nat, n : Nat, k : Nat]
```

$$\mathit{Send}(m) \triangleq \mathit{msgs}' = \mathit{msgs} \cup \{m\}$$

$$\mathit{SendAll}(\mathit{ms}) \triangleq \land\, \mathit{msgs}' = \mathit{msgs} \cup \mathit{ms}$$

$$\mathit{Statuses} \triangleq \{\mathit{Normal},\ \mathit{ViewChange},\ \mathit{Recovering}\}$$

$$\begin{array}{l}
\mathit{TypeOK} \triangleq \land\, \mathit{replicaState} \in [\\
\qquad\qquad \mathit{Replica} \rightarrow [\\
\qquad\qquad\qquad \mathit{viewNumber} : \mathit{Nat}, \\
\qquad\qquad\qquad \mathit{status} : \mathit{Statuses}, \\
\qquad\qquad\qquad \mathit{log} : \mathit{Seq}(\mathit{LogEntry}), \\
\qquad\qquad\qquad \mathit{downloadReplica} : \mathit{Replica} \cup \{\mathit{None}\}, \\
\qquad\qquad\qquad \mathit{commitNumber} : \mathit{Nat} \\
\qquad\qquad\quad] \\
\qquad\qquad] \\
\qquad\quad\ \land\, \mathit{msgs} \in \textsc{subset}\ \mathit{Message}
\end{array}$$

$$\begin{array}{ll}
\textsc{assume}\ \mathit{QuorumAssumption} \triangleq & \land\, \forall\, Q \in \mathit{Quorum} : Q \subseteq \mathit{Replica} \\
& \land\, \forall\, Q1,\ Q2 \in \mathit{Quorum} : Q1 \cap Q2 \neq \{\}
\end{array}$$

$$\textsc{assume}\ \mathit{IsFiniteSet}(\mathit{Replica})$$

$$\mathit{Max}(S) \triangleq \textsc{choose}\ x \in S : \forall\, y \in S : y \leq x$$

$$\mathit{Min}(S) \triangleq \textsc{choose}\ x \in S : \forall\, y \in S : x \leq y$$

$$\mathit{lastOpNumber}(l) \triangleq \textsc{if}\ l = \langle\rangle\ \textsc{then}\ 0\ \textsc{else}\ \ l[\mathit{Len}(l)].\mathit{opNumber}$$

$$\begin{array}{l}
\mathit{Init} \triangleq \land\, \mathit{replicaState} = [r \in \mathit{Replica} \mapsto [\\
\qquad\qquad\qquad\quad \mathit{viewNumber} \mapsto 0, \\
\qquad\qquad\qquad\quad \mathit{status} \mapsto \mathit{Normal}, \\
\qquad\qquad\qquad\quad \mathit{log} \mapsto \langle[\mathit{type} \mapsto \mathit{ViewBlock},\ \mathit{view} \mapsto 0]\rangle, \\
\qquad\qquad\qquad\quad \mathit{downloadReplica} \mapsto \mathit{None}, \\
\qquad\qquad\qquad\quad \mathit{commitNumber} \mapsto 0 \\
\qquad\qquad\qquad] \\
\qquad\qquad] \\
\qquad\quad \land\, \mathit{msgs} = \{\}
\end{array}$$

Getters

$$\mathit{ViewNumber}(r) \triangleq \mathit{replicaState}[r].\mathit{viewNumber}$$

$$\mathit{Status}(r) \triangleq \mathit{replicaState}[r].\mathit{status}$$

$$\mathit{Log}(r) \triangleq \mathit{replicaState}[r].\mathit{log}$$

$LogLen(r) \triangleq Len(Log(r))$

$LastNormalView(r) \triangleq Max(\{Log(r)[v].view : v \in \{i \in 1 \mathrel{..} LogLen(r) : Log(r)[i].type = ViewBlock\}\})$

$OpNumber(r) \triangleq LogLen(r)$

$DownloadReplica(r) \triangleq replicaState[r].downloadReplica$

$CommitNumber(r) \triangleq replicaState[r].commitNumber$

Helpful functions

$ReplicaIndex(r) \triangleq \textsc{choose}\ i \in 1 \mathrel{..} Cardinality(Replica) : ReplicaSequence[i] = r$

$PrimaryReplicaInView(v) \triangleq ReplicaSequence[(v\%Len(ReplicaSequence)) + 1]$

$IsPrimaryInView(r, v) \triangleq PrimaryReplicaInView(v) = r$

$IsPrimary(r) \triangleq IsPrimaryInView(r, ViewNumber(r))$

$$\begin{array}{l} IsDownloading(r) \triangleq \\ \quad \land replicaState[r].downloadReplica \neq None \end{array}$$

$RequestBlockCount(log) \triangleq Cardinality(\{i \in \textsc{domain}\ log : log[i].type = RequestBlock\})$

$ViewBlockCount(log) \triangleq Cardinality(\{i \in \textsc{domain}\ log : log[i].type = ViewBlock\})$

$$\begin{array}{l} AddClientRequest(r, m) \triangleq \\ \quad \land replicaState' = [replicaState\ \textsc{except}\ ![r].log = Append(@, m)] \end{array}$$

$$\begin{array}{l} RecieveClientRequest(p, op) \triangleq \\ \quad \land RequestBlockCount(Log(p)) < MaxRequests \\ \quad \land IsPrimary(p) \\ \quad \land Status(p) = Normal \\ \quad \land \lnot IsDownloading(p) \\ \quad \land AddClientRequest(p, [type \mapsto RequestBlock, \\ \qquad\qquad\qquad\qquad\qquad opNumber \mapsto OpNumber(p) + 1, \\ \qquad\qquad\qquad\qquad\qquad op \mapsto op]) \\ \quad \land Send([type \mapsto DownloadChunk, \\ \qquad\qquad v \mapsto ViewNumber(p), m \mapsto Log(p)'[OpNumber(p) + 1], \\ \qquad\qquad n \mapsto OpNumber(p) + 1, k \mapsto CommitNumber(p), i \mapsto p]) \end{array}$$

$$\begin{array}{l} RecievePrepare(r, m) \triangleq \\ \quad \land RequestBlockCount(Log(r)) < MaxRequests \\ \quad \land \lnot IsPrimary(r) \\ \quad \land Status(r) = Normal \\ \quad \land \lnot IsDownloading(r) \\ \quad \land m.type = DownloadChunk \end{array}$$

```
    $\land\ m.v = ViewNumber(r)$
    $\land\ m.n = OpNumber(r) + 1$
    $\land\ m.i\ = PrimaryReplicaInView(ViewNumber(r))$
    $\land\ AddClientRequest(r,\ m.m)$
    $\land\ Send([type \mapsto PrepareOk,$
            $v \mapsto ViewNumber(r),\ n \mapsto m.n,\ i \mapsto r])$

$PrepareOperation(r)\ \triangleq$
    $\land\ \neg IsPrimary(r)$
    $\land\ Status(r) = Normal$
    $\land\ \neg IsDownloading(r)$
    $\land\ \text{LET}\ maxPreparedOpNum\ \triangleq\ Max(\{0\} \cup \{m.n : m \in \{m \in msgs : m.type = PrepareOk \land m.i = r \land m.v$
      $\text{IN}\quad \land\ LogLen(r) > maxPreparedOpNum$
            $\land\ Send([type \mapsto PrepareOk,\ v \mapsto ViewNumber(r),$
                    $n \mapsto maxPreparedOpNum + 1,\ i \mapsto r])$
    $\land\ \text{UNCHANGED}\ \langle replicaState \rangle$

$AchievePrepareOkFromQuorum(p)\ \triangleq$
    $\land\ IsPrimary(p)$
    $\land\ Status(p) = Normal$
    $\land\ \neg IsDownloading(p)$
    $\land\ \text{LET}\ newCommit\ \triangleq\ CommitNumber(p) + 1$
      $\text{IN}\quad \land\ \exists\ Q \in Quorum :$
                $\forall\ r \in Q :$
                    $\lor\ Q \subseteq \{r\} \cup \{m.i : m \in \{m \in msgs : m.type = PrepareOk \land m.v = ViewNumber(p) \land m.n$
                    $\lor\ r = p$
            $\land\ replicaState' = [replicaState\ \text{EXCEPT}\ ![p].commitNumber = newCommit]$
            $\land\ Send([type \mapsto Commit,\ v \mapsto ViewNumber(p),\ k \mapsto replicaState[p].commitNumber'])$

$RecieveCommit(r,\ m)\ \triangleq$
    $\land\ \neg IsPrimary(r)$
    $\land\ Status(r) = Normal$
    $\land\ \neg IsDownloading(r)$
    $\land\ m.type = Commit$
    $\land\ m.v = ViewNumber(r)$
    $\land\ m.k > CommitNumber(r)$
    $\land\ replicaState' = [replicaState\ \text{EXCEPT}\ ![r].commitNumber = m.k]$
    $\land\ \text{UNCHANGED}\ \langle msgs \rangle$
```

View Changing

```
$TimeoutStartViewChanging(r)\ \triangleq$
    $\land\ ViewNumber(r) + 1 < MaxViews$
    $\land\ Status(r) = Normal$
    $\land\ replicaState' = [replicaState\ \text{EXCEPT}\ ![r].downloadReplica = None,$
```

```
                                          $![r].viewNumber = @ + 1,$
                                          $![r].status = ViewChange]$
   $\land\ Send([type \mapsto StartViewChange,\ v \mapsto ViewNumber(r)',\ i \mapsto r])$

$CheckAchieveStartViewChangeFromQuorum(r,\ v)\ \triangleq$
   $\land$ IF $\exists\, Q \in Quorum : \land\ r\ \in Q$
                                  $\land\ Q = \{r\} \cup \{m.i : m \in \{mm \in msgs : mm.type = StartViewChange$
                                                                   $\land\quad mm.v = replicaState'[r].viewNumber\}\}$
      THEN $Send([type \mapsto DoViewChange,\ v \mapsto v,\ vv \mapsto LastNormalView(r),$
                 $n \mapsto OpNumber(r),\ k \mapsto CommitNumber(r),\ i \mapsto r])$
      ELSE UNCHANGED $\langle msgs \rangle$

$RecieveStartViewChange(r,\ m)\ \triangleq$
   $\land\ m.type = StartViewChange$
   $\land\ \lor$ Start View Changing
       $\land\ ViewNumber(r) < m.v$
       $\land\ replicaState'\ = [replicaState$ EXCEPT $![r].downloadReplica = None,$
                                              $![r].viewNumber = m.v,$
                                              $![r].status = ViewChange]$
       $\land\ CheckAchieveStartViewChangeFromQuorum(r,\ m.v)$
     $\lor$ Our view number
       $\land\ ViewNumber(r) = m.v$
       $\land\ Status(r) = ViewChange$
       $\land\ \exists\, Q \in Quorum : \land\ r \in Q$
                                $\land\ Q \subseteq \{r\} \cup \{mm.i : mm \in \{mmm \in msgs : mmm.type = StartViewChange$
                                                                $\land\quad mmm.v = m.v\}\}$
       $\land\ Send([type \mapsto DoViewChange,\ v \mapsto m.v,\ vv \mapsto LastNormalView(r),$
                $n \mapsto OpNumber(r),\ k \mapsto CommitNumber(r),\ i \mapsto r])$
       $\land$ UNCHANGED $\langle replicaState \rangle$

$RecieveDoViewChange(p,\ m)\ \triangleq$
   $\land\ m.type = DoViewChange$
   $\land\ IsPrimaryInView(p,\ m.v)$
   $\land\ ViewNumber(p) < m.v$
   $\land\ replicaState'\ = [replicaState$ EXCEPT $![p].downloadReplica = None,$
                                          $![p].viewNumber = m.v,$
                                          $![p].status = ViewChange]$
   $\land$ UNCHANGED $\langle msgs \rangle$

Become Primary
$AchieveDoViewChangeFromQuorum(p)\ \triangleq$
   $\land\ IsPrimary(p)$
   $\land\ Status(p) = ViewChange$
   $\land$ LET $recieved\ \triangleq\ \{m \in msgs : m.type = DoViewChange \land m.v = ViewNumber(p)\} \cup$
                        $\{[type \mapsto DoViewChange,\ v \mapsto ViewNumber(p),\ vv \mapsto LastNormalView(p),$
                          $n \mapsto OpNumber(p),\ k \mapsto CommitNumber(p),\ i \mapsto p]\}$
```

```
    IN   ∧ ∃ Q ∈ Quorum : ∧ p ∈ Q
                           ∧ Q ⊆ {m.i : m ∈ recieved}

         ∧ LET maxVV ≜ Max({m.vv : m ∈ recieved})
               maxN ≜ Max({m.n : m ∈ {m ∈ recieved : m.vv = maxVV}})
               maxReplicaIndex ≜ Max({ReplicaIndex(m.i) : m ∈ {m ∈ recieved : m.vv = maxVV ∧ m.n
               maxReplica ≜ IF ∧ maxVV = LastNormalView(p)
                               ∧ maxN = OpNumber(p)
                            THEN p
                            ELSE (CHOOSE m ∈ recieved : ReplicaIndex(m.i) = maxReplicaIndex).i
           IN  ∧ replicaState′ = [replicaState EXCEPT ![p].downloadReplica = IF maxReplica = p
                                                                               THEN None
                                                                               ELSE maxReplica,
                                                      ![p].log = IF maxReplica = p
                                                                    THEN Append(@, [type ↦ ViewBlock,
                                                                    ELSE @,
                                                      ![p].status = Normal]
               ∧ IF maxReplica = p
                    THEN Send([type ↦ StartView, v ↦ ViewNumber(p), n ↦ OpNumber(p)′, k ↦ replicaS
                    ELSE Send([type ↦ StartDownload, v ↦ ViewNumber(p), n ↦ CommitNumber(p) + 1,

SendDownloadChunks(r) ≜
    ∧ Status(r) ≠ Recovering
    ∧ ∃ m ∈ msgs :
        ∧ m.type = StartDownload
        ∧ m.src = r
        ∧ m.v = ViewNumber(r)
        ∧ SendAll({[type ↦ DownloadChunk,
                    v ↦ ViewNumber(r), m ↦ Log(r)[opNum],
                    n ↦ opNum, k ↦ CommitNumber(r), i ↦ r] : opNum ∈ m.n .. LogLen(r)})
        ∧ UNCHANGED ⟨replicaState⟩

 Mc → Mc / Mc → M
MasterDownloadBeforeView(p) ≜
    ∧ IsPrimary(p)
    ∧ Status(p) ≠ Recovering
    ∧ IsDownloading(p)
    ∧ LET msgsToDownload ≜ {msg ∈ msgs :
                              ∧ msg.type = DownloadChunk
                              ∧ msg.v = ViewNumber(p)
                              ∧ msg.i = DownloadReplica(p)
                              ∧ ∨ LogLen(p) + 1 = msg.n
                                ∨ ∧ LogLen(p) ≥ msg.n
                                  ∧ Log(p)[msg.n] ≠ msg.m}
      IN  ∧ msgsToDownload ≠ {}
          ∧ LET doViewChangeReceived ≜ {m ∈ msgs : m.type = DoViewChange ∧ m.v = ViewNumber(p)
      $MinOpNum \triangleq Min(\{msg.n : msg \in msgsToDownload\})$
      $MinMsg \triangleq \textsc{choose}\ msg \in msgsToDownload : msg.n = MinOpNum$
      $finished \triangleq MinOpNum \ \geq (\textsc{choose}\ m \in doViewChangeReceived : m.i = DownloadReplica$
      $\textsc{in}$  $\land\ replicaState' = [replicaState\ \textsc{except}\ ![p].log =$
                                                  $\textsc{if}\ finished$
                                                   $\textsc{then}\ Append($
                                                          $Append(SubSeq(@, 1, MinOpNu$
                                                          $[type \mapsto ViewBlock, view \mapsto Vie$
                                                      $)$
                                                   $\textsc{else}\ Append(SubSeq(@, 1, MinOpNum -$
                                               $![p].downloadReplica =$
                                                  $\textsc{if}\ finished$
                                                   $\textsc{then}\ None$
                                                   $\textsc{else}\ @]$
         $\land\ \textsc{if}\ finished$
            $\textsc{then}\ Send([type \mapsto StartView,$
                          $v \mapsto ViewNumber(p),$
                          $n \mapsto OpNumber(p)',$
                          $k \mapsto replicaState[p].commitNumber'])$
            $\textsc{else}\ \ \textsc{unchanged}\ \langle msgs \rangle$

$RecieveStartView(r, m) \triangleq$
   $\land\ m.type = StartView$
   $\land\ \lor\ ViewNumber(r) < m.v$
      $\lor\ \land\ ViewNumber(r) = m.v$
         $\land\ Status(r) = ViewChange$
   $\land\ replicaState' = [replicaState\ \textsc{except}\ ![r].downloadReplica = PrimaryReplicaInView(m.v),$
                                              $![r].viewNumber = m.v,$
                                              $![r].status = Normal]$
   $\land\ Send([type \mapsto StartDownload, v \mapsto m.v, n \mapsto CommitNumber(r) + 1, src \mapsto PrimaryReplicaInView(m.$

 $Rc \rightarrow Rc\ /\ Rc \rightarrow R$
$ReplicaDownloadBeforeView(r) \triangleq$
   $\land\ \neg IsPrimary(r)$
   $\land\ Status(r) = Normal$
   $\land\ IsDownloading(r)$
   $\land\ \textsc{let}\ allMsgsToDownload \triangleq \{msg \in msgs :$
                                       $\land\ msg.type = DownloadChunk$
                                       $\land\ msg.v = ViewNumber(r)$
                                       $\land\ msg.n > CommitNumber(r)$
                                       $\land\ msg.i = DownloadReplica(r)\}$
         $msgsToDownload \triangleq \{msg \in allMsgsToDownload :$
                                       $\lor\ LogLen(r) + 1 = msg.n$
                                       $\lor\ \land\ LogLen(r) \geq msg.n$
                                          $\land\ Log(r)[msg.n] \neq msg.m\}$
```

```
    IN  $\land\ msgsToDownload \neq \{\}$
        $\land$ LET $MinOpNum \triangleq Min(\{msg.n : msg \in msgsToDownload\})$
                $MinMsg \triangleq$ CHOOSE $msg \in msgsToDownload : msg.n = MinOpNum$
                $finished \triangleq MinMsg.m = [type \mapsto ViewBlock,\ view \mapsto ViewNumber(r)]$
            IN  $\land$  all previous chunks are exist (or else it is another download)
                  $\forall\, prevPos \in CommitNumber(r) + 1 \,..\, MinMsg.n - 1 :$
                     $\exists\, prev \in allMsgsToDownload :$
                         $\land\ prev.type = DownloadChunk$
                         $\land\ prev.v = ViewNumber(r)$
                         $\land\ prev.i = DownloadReplica(r)$
                         $\land\ prev.n = prevPos$
                $\land\ replicaState' = [replicaState$ EXCEPT $![r].log = Append(SubSeq(@,\ 1,\ MinOpNum - 1),\ M$
                                                     $![r].downloadReplica =$
                                                         IF $finished$
                                                          THEN $None$
                                                          ELSE $@]$
    $\land$ UNCHANGED $\langle msgs \rangle$
```

```
Finishing $\triangleq$
    $\land$ LET $r \triangleq ReplicaSequence[1]$
            All Committed
        IN  $\land\ CommitNumber(r) = OpNumber(r)$
            $MaxRequests$ commands are stored
            $\land\ RequestBlockCount(Log(r)) = MaxRequests$
            All replicas equal
            $\land\ \forall\, r1 \in Replica :$
                 $\land\ Log(r1) = Log(r)$
                 $\land\ CommitNumber(r1) = CommitNumber(r)$
                 $\land\ ViewNumber(r1) = ViewNumber(r)$
                 $\land\ Status(r1) = Normal$
                 $\land\ DownloadReplica(r1) = None$
    $\land$ UNCHANGED $\langle replicaState,\ msgs \rangle$
```

```
NormalOperationProtocol $\triangleq$
    $\lor\ \exists\, r \in Replica,\ op \in Operation : RecieveClientRequest(r,\ op)$
    $\lor\ \exists\, r \in Replica,\ m \in msgs : RecievePrepare(r,\ m)$
    $\lor\ \exists\, r \in Replica : PrepareOperation(r)$
    $\lor\ \exists\, p \in Replica : AchievePrepareOkFromQuorum(p)$
    $\lor\ \exists\, r \in Replica,\ m \in msgs : RecieveCommit(r,\ m)$

ViewChangeProtocol $\triangleq$
    $\lor\ \exists\, r \in Replica : TimeoutStartViewChanging(r)$
    $\lor \exists\, r \in \mathit{Replica},\ m \in \mathit{msgs} : \mathit{RecieveStartViewChange}(r,\ m)$
    $\lor \exists\, p \in \mathit{Replica},\ m \in \mathit{msgs} : \mathit{RecieveDoViewChange}(p,\ m)$
    $\lor \exists\, r \in \mathit{Replica} : \mathit{AchieveDoViewChangeFromQuorum}(r)$
    $\lor \exists\, r \in \mathit{Replica} : \mathit{SendDownloadChunks}(r)$
    $\lor \exists\, p \in \mathit{Replica} : \mathit{MasterDownloadBeforeView}(p)$
    $\lor \exists\, r \in \mathit{Replica},\ m \in \mathit{msgs} : \mathit{RecieveStartView}(r,\ m)$
    $\lor \exists\, r \in \mathit{Replica} : \mathit{ReplicaDownloadBeforeView}(r)$

$\mathit{Next} \triangleq$ $\lor\ \mathit{NormalOperationProtocol}$
         $\lor\ \mathit{ViewChangeProtocol}$
         $\lor\ \mathit{Finishing}$
```

Full *Spec*

```
$\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}}$

$\mathit{FullSpec} \triangleq$ $\land\ \mathit{Init}$
             $\land\ \Box[\mathit{Next}]_{\mathit{vars}}$
             $\land\ \mathrm{WF}_{\langle \mathit{vars} \rangle}(\mathit{Next})$
```

```
$\mathit{VRNoMsgs} \triangleq \textsc{instance}\ \mathit{VR\_without\_message}$

$\textsc{theorem}\ \mathit{Spec} \Rightarrow \mathit{VRNoMsgs}!\mathit{Spec}$
```

INVARIANTS

```
$\mathit{CommitedLogsPreficesAreEqual} \triangleq$
    $\forall\, r1,\ r2 \in \mathit{Replica} :$
       $\forall\, i \in \textsc{domain}\ \mathit{Log}(r1)$
           $\cap\ \textsc{domain}\ \mathit{Log}(r2)$
           $\cap\ 1 \,..\, \mathit{Min}(\{\mathit{CommitNumber}(r1),$
                           $\mathit{CommitNumber}(r2)\}) :$
             $\mathit{Log}(r1)[i] = \mathit{Log}(r2)[i]$

$\mathit{KeepMaxRequests} \triangleq \forall\, r \in \mathit{Replica} : \mathit{RequestBlockCount}(\mathit{Log}(r)) \leq \mathit{MaxRequests}$

$\mathit{KeepMaxViews} \triangleq \forall\, r \in \mathit{Replica} : \mathit{ViewNumber}(r) + 1 \leq \mathit{MaxViews}$
```

Properties

```
$\mathit{EventuallyFinished} \triangleq \Diamond(\textsc{enabled}\ \mathit{Finishing})$
```

* Modification History
* Last modified *Mon* May 15 09:33:48 *MSK* 2023 by *tycoon*
* Created *Mon Nov* 07 20:04:34 *MSK* 2022 by *tycoon*