MODULE *Agent*

EXTENDS *FiniteSets*, *Naturals*, *Sequences*, *TLC*

CONSTANTS *Emails* — Set of incoming *Emails*

VARIABLES
Archived,
Arrived, — Queue of incoming *Emails*
Completed, — Queue of completion responses
RemoteOutbox, — Set of outgoing *Emails*
Parsed, — Set of parsed *Emails*
Sending,
Abandoned — Set of failed *Emails*

$vars \triangleq \langle Abandoned, Archived, Arrived, Completed, Parsed, RemoteOutbox, Sending \rangle$

$EmailsInQueue \triangleq Abandoned \cup Archived \cup Arrived \cup Completed \cup Parsed$

$$
\begin{aligned}
TypeOK \triangleq\; & \land\ Abandoned \subseteq Emails \\
& \land\ Archived \subseteq Emails \\
& \land\ Arrived \subseteq Emails \\
& \land\ Completed \subseteq Emails \\
& \land\ Parsed \subseteq Emails \\
& \land\ RemoteOutbox \in Seq(Emails) \\
& \land\ Sending \subseteq Emails
\end{aligned}
$$

$Range(S) \triangleq \{S[n] : n \in \text{DOMAIN } S\}$

$Invariants \triangleq$

Don’t parse e-mails more than once.

$\land\ \forall\, email \in Completed : email \notin Parsed \Rightarrow email \notin Arrived$

Abandoned e-mails not to appear anywhere else, as *Abandoned* is a general queue state separate from e-mail processing state.

$\land\ \forall\, email \in Abandoned : email \notin Arrived \cup Completed \cup Parsed$

Don’t send e-mails more than once.

$\land\ Len(RemoteOutbox) = Cardinality(Range(RemoteOutbox))$

$ReceiveEmailOK(email) \triangleq$

Enqueues an *Email* from *Inbox* to *Arrived*.

$\land\ Arrived' = Arrived \cup \{email\}$
$\land\ \text{UNCHANGED } \langle Abandoned, Archived, Completed, Parsed, RemoteOutbox, Sending \rangle$

$ReceiveEmailError(email) \triangleq$

Fails reading an *email* from *Inbox*. Logs it, marks it and moves it to *RemoteArchived* folder. Support engineer can move the *email* back to *Inbox* after addressing the issue.

$\land\ Abandoned' = Abandoned \cup \{email\}$
$\land\ \text{UNCHANGED } \langle Archived, Arrived, Completed, Parsed, RemoteOutbox, Sending \rangle$

$ReceiveEmail \triangleq\ \land\ \exists\, email \in Emails \setminus EmailsInQueue :$

$\lor\ ReceiveEmailOK(email)$
$\lor\ ReceiveEmailError(email)$

$ParseEmail1OK(email) \triangleq$

The first step of parsing an e-mail response stores the parsed content in the queue.

$\land\ email \notin Parsed$
$\land\ Parsed' = Parsed \cup \{email\}$
$\land$ UNCHANGED $\langle Abandoned, Archived, Arrived, Completed, RemoteOutbox, Sending \rangle$

$ParseEmail2OK(email) \triangleq$

The second step of parsing removes the e-mail response from the queue only after the parsing is successful. This ensures we don't lose any e-mails in case of a failure.

$\land\ email \in Parsed$
$\land\ Arrived' = Arrived \setminus \{email\}$
$\land$ UNCHANGED $\langle Abandoned, Archived, Completed, Parsed, RemoteOutbox, Sending \rangle$

$ParseEmailOK(email) \triangleq$

Parses an *email*. The sub-operations occur over distributed settings and may fail. Each sub-operation is atomic, and their order of execution is important.

$\lor\ ParseEmail1OK(email)$
$\lor\ ParseEmail2OK(email)$

$ParseEmail1Error(email) \triangleq$

Fails parsing an *email*.

$\land\ email \notin Parsed$
$\land\ Abandoned' = Abandoned \cup \{email\}$
$\land\ Arrived' = Arrived \setminus \{email\}$
$\land$ UNCHANGED $\langle Archived, Completed, Parsed, RemoteOutbox, Sending \rangle$

$ParseEmail \triangleq$

```
∃ email ∈ Arrived \ Abandoned :
    ∨ ParseEmailOK(email)
    ∨ ParseEmail1Error(email)
```

$CompleteMessage1OK(email) \triangleq$

$\land\ email \notin Completed$
$\land\ Completed' = Completed \cup \{email\}$
$\land$ UNCHANGED $\langle Abandoned, Archived, Arrived, Parsed, RemoteOutbox, Sending \rangle$

$CompleteMessage2OK(email) \triangleq$

$\land\ email \in Completed$
$\land\ Parsed' = Parsed \setminus \{email\}$
$\land$ UNCHANGED $\langle Abandoned, Archived, Arrived, Completed, RemoteOutbox, Sending \rangle$

$CompleteMessageOK(email) \triangleq$

$\lor\ CompleteMessage1OK(email)$

$\lor\ CompleteMessage2OK(email)$

$CompleteMessage1Error(email)\ \triangleq$
$\quad \land\ email \notin Completed$
$\quad \land\ Abandoned' = Abandoned \cup \{email\}$
$\quad \land\ Parsed' = Parsed \setminus \{email\}$
$\quad \land\ \text{UNCHANGED}\ \langle Archived, Arrived, Completed, RemoteOutbox, Sending\rangle$

$CompleteMessage\ \triangleq$
$\quad \exists\, email \in Parsed \setminus (Arrived \cup Abandoned):$
$\qquad \lor\ CompleteMessageOK(email)$
$\qquad \lor\ CompleteMessage1Error(email)$

$SendOutCompletion1OK\ \triangleq$

Sends out a completion response e-mail.

$\quad \exists\, email \in Completed \setminus (Abandoned \cup Parsed):$
$\qquad \land\ email \notin Sending$
$\qquad \land\ Sending' = \{email\}$
$\qquad \land\ RemoteOutbox' = Append(RemoteOutbox, email)$
$\qquad \land\ \text{UNCHANGED}\ \langle Abandoned, Archived, Arrived, Completed, Parsed\rangle$

$SendOutCompletion2OK\ \triangleq$

Marks an *email* as sent.

$\quad \land\ Sending \neq \{\}$
$\quad \land\ Sending' = \{\}$
$\quad \land\ Completed' = Completed \setminus Sending$
$\quad \land\ Archived' = Archived \cup Sending$
$\quad \land\ \text{UNCHANGED}\ \langle Abandoned, Arrived, Parsed, RemoteOutbox\rangle$

$SendOutCompletion2Error\ \triangleq$

Fails marking an *email* as sent.

$\quad \land\ Sending \neq \{\}$
$\quad \land\ Sending' = \{\}$
$\quad \land\ \text{UNCHANGED}\ \langle Abandoned, Archived, Arrived, Completed, Parsed, RemoteOutbox\rangle$

$SendOutCompletion\ \triangleq$
$\quad \lor\ SendOutCompletion1OK$
$\quad \lor\ SendOutCompletion2OK$
$\quad \lor\ SendOutCompletion2Error$

$AllDone\ \triangleq$

All done and system comes to equilibrium.

$\quad \land\ Archived \cup Abandoned = Emails$
$\quad \land\ Parsed \setminus Abandoned = \{\}$
$\quad \land\ \text{UNCHANGED}\ vars$

$$Init \triangleq \land Abandoned = \{\} \\ \land Archived = \{\} \\ \land Arrived = \{\} \\ \land Completed = \{\} \\ \land Parsed = \{\} \\ \land RemoteOutbox = \langle\rangle \\ \land Sending = \{\}$$

$$Next \triangleq \lor ReceiveEmail \\ \lor ParseEmail \\ \lor CompleteMessage \\ \lor SendOutCompletion \\ \lor AllDone$$

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

Temporal properties for verification

$NoLostEmails \triangleq$

No e-mails should be lost. This is a safety property.

$$\forall\, email \in Emails : \\ \Box(email \in EmailsInQueue \Rightarrow \Diamond\Box(email \in Abandoned \cup Archived))$$

THEOREM $Spec \Rightarrow \Box\, TypeOK$
THEOREM $Spec \Rightarrow \Box Invariants$
THEOREM $Spec \Rightarrow NoLostEmails$

\* Modification History
\* Last modified *Tue* May 02 16:13:11 *KST* 2023 by *hcs*
\* Created *Fri Apr* 28 13:04:37 *KST* 2023 by *hcs*