MODULE *Agent*

EXTENDS *TLC*

CONSTANTS	$Emails$	Set of incoming *Emails*
VARIABLES	$Arrived$,	Queue of incoming *Emails*
	$Completed$,	Queue of completion responses
	$Outbox$,	Set of outgoing *Emails*
	$Parsed$,	Set of parsed *Emails*
	$Abandoned$	Set of failed *Emails*

$vars \triangleq \langle Abandoned, Arrived, Completed, Outbox, Parsed \rangle$

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

$$
\begin{aligned}
TypeOK \triangleq\ & \land Abandoned \subseteq Emails \\
& \land Arrived \subseteq Emails \\
& \land Completed \subseteq Emails \\
& \land Outbox \subseteq Emails \\
& \land Parsed \subseteq Emails
\end{aligned}
$$

$Invariants \triangleq$

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.

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

$ReceiveEmailOK(email) \triangleq$

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

$$
\begin{aligned}
& \land Arrived' = Arrived \cup \{email\} \\
& \land \text{UNCHANGED } \langle Abandoned, Completed, Outbox, Parsed \rangle
\end{aligned}
$$

$ReceiveEmailError(email) \triangleq$

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.

$$
\begin{aligned}
& \land Abandoned' = Abandoned \cup \{email\} \\
& \land \text{UNCHANGED } \langle Arrived, Completed, Outbox, Parsed \rangle
\end{aligned}
$$

$$
\begin{aligned}
ReceiveEmail \triangleq\ & \land \exists\, email \in Emails \setminus EmailsInQueue : \\
& \qquad \lor ReceiveEmailOK(email) \\
& \qquad \lor ReceiveEmailError(email)
\end{aligned}
$$

$ParseEmail1OK(email) \triangleq$

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

$$
\begin{aligned}
& \land email \notin Parsed \\
& \land Parsed' = Parsed \cup \{email\} \\
& \land \text{UNCHANGED } \langle Abandoned, Arrived, Completed, Outbox \rangle
\end{aligned}
$$

$ParseEmail2OK(email) \triangleq$

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.

$$\begin{aligned}
&\land\ email \in Parsed\\
&\land\ Arrived' = Arrived \setminus \{email\}\\
&\land\ \text{UNCHANGED}\ \langle Abandoned,\ Completed,\ Outbox,\ Parsed\rangle
\end{aligned}$$

$ParseEmailOK(email) \triangleq$

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.

$$\begin{aligned}
&\lor\ ParseEmail1OK(email)\\
&\lor\ ParseEmail2OK(email)
\end{aligned}$$

$ParseEmail1Error(email) \triangleq$

Fails parsing an *email*.

$$\begin{aligned}
&\land\ email \notin Parsed\\
&\land\ Abandoned' = Abandoned \cup \{email\}\\
&\land\ Arrived' = Arrived \setminus \{email\}\\
&\land\ \text{UNCHANGED}\ \langle Completed,\ Outbox,\ Parsed\rangle
\end{aligned}$$

$ParseEmail \triangleq$

$$\begin{aligned}
&\exists\, email \in Arrived \setminus Abandoned :\\
&\qquad \lor\ ParseEmailOK(email)\\
&\qquad \lor\ ParseEmail1Error(email)
\end{aligned}$$

$CompleteMessage1OK(email) \triangleq$

$$\begin{aligned}
&\land\ email \notin Completed\\
&\land\ Completed' = Completed \cup \{email\}\\
&\land\ \text{UNCHANGED}\ \langle Abandoned,\ Arrived,\ Outbox,\ Parsed\rangle
\end{aligned}$$

$CompleteMessage2OK(email) \triangleq$

$$\begin{aligned}
&\land\ email \in Completed\\
&\land\ Parsed' = Parsed \setminus \{email\}\\
&\land\ \text{UNCHANGED}\ \langle Abandoned,\ Arrived,\ Completed,\ Outbox\rangle
\end{aligned}$$

$CompleteMessageOK(email) \triangleq$

$$\begin{aligned}
&\lor\ CompleteMessage1OK(email)\\
&\lor\ CompleteMessage2OK(email)
\end{aligned}$$

$CompleteMessage1Error(email) \triangleq$

$$\begin{aligned}
&\land\ email \notin Completed\\
&\land\ Abandoned' = Abandoned \cup \{email\}\\
&\land\ Parsed' = Parsed \setminus \{email\}\\
&\land\ \text{UNCHANGED}\ \langle Arrived,\ Completed,\ Outbox\rangle
\end{aligned}$$

$CompleteMessage \triangleq$

$$\exists\, email \in Parsed \setminus (Arrived \cup Abandoned) :$$

$$\begin{aligned}
&\lor\ \mathit{CompleteMessageOK}(\mathit{email})\\
&\lor\ \mathit{CompleteMessage1Error}(\mathit{email})
\end{aligned}$$

$$\begin{aligned}
&\mathit{AllEmailsCompletedOrUndeliverable}\ \triangleq\\
&\qquad \land\ \mathit{Completed} \cup \mathit{Abandoned} = \mathit{Emails}\\
&\qquad \land\ \mathit{Parsed} \setminus \mathit{Abandoned} = \{\}\\
&\qquad \land\ \textsc{unchanged}\ \mathit{vars}
\end{aligned}$$

$$\begin{aligned}
\mathit{Init}\ \triangleq\ &\land\ \mathit{Abandoned} = \{\}\\
&\land\ \mathit{Arrived} = \{\}\\
&\land\ \mathit{Completed} = \{\}\\
&\land\ \mathit{Outbox} = \{\}\\
&\land\ \mathit{Parsed}\ = \{\}
\end{aligned}$$

$$\begin{aligned}
\mathit{Next}\ \triangleq\ &\lor\ \mathit{ReceiveEmail}\\
&\lor\ \mathit{ParseEmail}\\
&\lor\ \mathit{CompleteMessage}\\
&\lor\ \mathit{AllEmailsCompletedOrUndeliverable}
\end{aligned}$$

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

Temporal properties for verification

$$\mathit{NoLostEmails}\ \triangleq$$

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

$$\begin{aligned}
&\forall\, \mathit{email} \in \mathit{Emails} :\\
&\qquad \Box(\mathit{email} \in \mathit{EmailsInQueue} \Rightarrow \Diamond\Box(\mathit{email} \in \mathit{Abandoned} \cup \mathit{Completed}))
\end{aligned}$$

$\textsc{theorem}\ \mathit{Spec} \Rightarrow \Box\,\mathit{TypeOK}$

$\textsc{theorem}\ \mathit{Spec} \Rightarrow \Box\,\mathit{Invariants}$

$\textsc{theorem}\ \mathit{Spec} \Rightarrow \mathit{NoLostEmails}$

---

* Modification History
* Last modified *Tue* May 02 11:52:01 *KST* 2023 by *hcs*
* Created *Fri Apr* 28 13:04:37 *KST* 2023 by *hcs*