MODULE *Agent*

Reliably Interface with *GPT* via *Gmail*

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

CONSTANTS *Emails* — Set of incoming *Emails*

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

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

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

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

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

$Invariants \triangleq$

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

Don't parse e-mails more than once.

$\land \forall email \in Range(RemoteOutbox) : email \notin Completed \Rightarrow email \notin Parsed$

Don't complete e-mails more than once.

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

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

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

Same with archived emails.

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

Don't send e-mails more than once.

$\mathit{ReceiveEmailOK}(\mathit{email}) \triangleq$

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

$$\begin{aligned} &\land \mathit{Arrived}' = \mathit{Arrived} \cup \{\mathit{email}\} \\ &\land \textsc{unchanged}\ \langle \mathit{Abandoned}, \mathit{Archived}, \mathit{Completed}, \mathit{Parsed}, \mathit{RemoteOutbox} \rangle \end{aligned}$$

$\mathit{ReceiveEmailError}(\mathit{email}) \triangleq$

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.

$$\begin{aligned} &\land \mathit{Abandoned}' = \mathit{Abandoned} \cup \{\mathit{email}\} \\ &\land \textsc{unchanged}\ \langle \mathit{Archived}, \mathit{Arrived}, \mathit{Completed}, \mathit{Parsed}, \mathit{RemoteOutbox} \rangle \end{aligned}$$

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

$\mathit{ParseEmail1OK}(\mathit{email}) \triangleq$

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

$$\begin{aligned} &\land \mathit{email} \notin \mathit{Parsed} \\ &\land \mathit{Parsed}' = \mathit{Parsed} \cup \{\mathit{email}\} \\ &\land \textsc{unchanged}\ \langle \mathit{Abandoned}, \mathit{Archived}, \mathit{Arrived}, \mathit{Completed}, \mathit{RemoteOutbox} \rangle \end{aligned}$$

$\mathit{ParseEmail2OK}(\mathit{email}) \triangleq$

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.

$$\begin{aligned} &\land \mathit{email} \in \mathit{Parsed} \\ &\land \mathit{Arrived}' = \mathit{Arrived} \setminus \{\mathit{email}\} \\ &\land \textsc{unchanged}\ \langle \mathit{Abandoned}, \mathit{Archived}, \mathit{Completed}, \mathit{Parsed}, \mathit{RemoteOutbox} \rangle \end{aligned}$$

$\mathit{ParseEmailOK}(\mathit{email}) \triangleq$

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.

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

$\mathit{ParseEmail1Error}(\mathit{email}) \triangleq$

Fails parsing an *email*.

$$\begin{aligned} &\land \mathit{email} \notin \mathit{Parsed} \\ &\land \mathit{Abandoned}' = \mathit{Abandoned} \cup \{\mathit{email}\} \\ &\land \mathit{Arrived}' = \mathit{Arrived} \setminus \{\mathit{email}\} \\ &\land \textsc{unchanged}\ \langle \mathit{Archived}, \mathit{Completed}, \mathit{Parsed}, \mathit{RemoteOutbox} \rangle \end{aligned}$$

$\mathit{ParseEmail} \triangleq$

$\quad \exists\, \mathit{email} \in \mathit{Arrived} \setminus \mathit{Abandoned} :$

$$\begin{array}{l}
\quad\quad \lor\ \mathit{ParseEmailOK}(\mathit{email}) \\
\quad\quad \lor\ \mathit{ParseEmail1Error}(\mathit{email})
\end{array}$$

$$\begin{array}{l}
\mathit{CompleteMessage1OK}(\mathit{email})\ \triangleq \\
\quad \land\ \mathit{email} \notin \mathit{Completed} \\
\quad \land\ \mathit{Completed}' = \mathit{Completed} \cup \{\mathit{email}\} \\
\quad \land\ \textsc{unchanged}\ \langle \mathit{Abandoned},\ \mathit{Archived},\ \mathit{Arrived},\ \mathit{Parsed},\ \mathit{RemoteOutbox} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{CompleteMessage2OK}(\mathit{email})\ \triangleq \\
\quad \land\ \mathit{email} \in \mathit{Completed} \\
\quad \land\ \mathit{Parsed}' = \mathit{Parsed} \setminus \{\mathit{email}\} \\
\quad \land\ \textsc{unchanged}\ \langle \mathit{Abandoned},\ \mathit{Archived},\ \mathit{Arrived},\ \mathit{Completed},\ \mathit{RemoteOutbox} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{CompleteMessageOK}(\mathit{email})\ \triangleq \\
\quad \lor\ \mathit{CompleteMessage1OK}(\mathit{email}) \\
\quad \lor\ \mathit{CompleteMessage2OK}(\mathit{email})
\end{array}$$

$$\begin{array}{l}
\mathit{CompleteMessage1Error}(\mathit{email})\ \triangleq \\
\quad \land\ \mathit{email} \notin \mathit{Completed} \\
\quad \land\ \mathit{Abandoned}' = \mathit{Abandoned} \cup \{\mathit{email}\} \\
\quad \land\ \mathit{Parsed}' = \mathit{Parsed} \setminus \{\mathit{email}\} \\
\quad \land\ \textsc{unchanged}\ \langle \mathit{Archived},\ \mathit{Arrived},\ \mathit{Completed},\ \mathit{RemoteOutbox} \rangle
\end{array}$$

$$\begin{array}{l}
\mathit{CompleteMessage}\ \triangleq \\
\quad \exists\, \mathit{email} \in \mathit{Parsed} \setminus (\mathit{Arrived} \cup \mathit{Abandoned}) : \\
\quad\quad \lor\ \mathit{CompleteMessageOK}(\mathit{email}) \\
\quad\quad \lor\ \mathit{CompleteMessage1Error}(\mathit{email})
\end{array}$$

$\mathit{SendOutCompletion1OK}(\mathit{email})\ \triangleq$

Sends out a completion response e-mail.

$$\begin{array}{ll}
\land\ \mathit{email} \notin \mathit{Range}(\mathit{RemoteOutbox}) & \text{We haven't already sent this e-mail} \\
\land\ \mathit{RemoteOutbox}' = \mathit{Append}(\mathit{RemoteOutbox},\ \mathit{email}) & \\
\land\ \textsc{unchanged}\ \langle \mathit{Abandoned},\ \mathit{Archived},\ \mathit{Arrived},\ \mathit{Completed},\ \mathit{Parsed} \rangle &
\end{array}$$

$\mathit{SendOutCompletion2OK}(\mathit{email})\ \triangleq$

Marks an *email* as sent.

$$\begin{array}{ll}
\land\ \mathit{email} \in \mathit{Range}(\mathit{RemoteOutbox}) & \text{Previous step to send this e-mail succeeded.} \\
\land\ \mathit{Archived}' = \mathit{Archived} \cup \{\mathit{email}\} & \\
\land\ \mathit{Completed}' = \mathit{Completed} \setminus \{\mathit{email}\} & \\
\land\ \textsc{unchanged}\ \langle \mathit{Abandoned},\ \mathit{Arrived},\ \mathit{Parsed},\ \mathit{RemoteOutbox} \rangle &
\end{array}$$

$\mathit{SendOutCompletion1Error}(\mathit{email})\ \triangleq$

Fails sending the e-mail.

$$\begin{array}{ll}
\land\ \mathit{email} \notin \mathit{Range}(\mathit{RemoteOutbox}) & \text{We haven't already sent this e-mail} \\
\land\ \mathit{Abandoned}' = \mathit{Abandoned} \cup \{\mathit{email}\} & \\
\land\ \mathit{Completed}' = \mathit{Completed} \setminus \{\mathit{email}\} &
\end{array}$$

```
    $\land$ UNCHANGED $\langle Archived, Arrived, Parsed, RemoteOutbox \rangle$

$SendOutCompletion \triangleq$
    $\exists\, email \in Completed \setminus (Abandoned \cup Parsed) :$
        $\lor\, SendOutCompletion1OK(email)$
        $\lor\, SendOutCompletion2OK(email)$
        $\lor\, SendOutCompletion1Error(email)$

$AllDone \triangleq$
    All done and system comes to equilibrium.
    $\land\, Archived \cup Abandoned = Emails$
    $\land\, Parsed \setminus Abandoned = \{\}$
    $\land$ UNCHANGED $vars$

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

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

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

Temporal properties for verification

$NoLostEmails \triangleq$
    No e-mails should be lost. This is a safety property.
    $\forall\, email \in Emails :$
        $\Box(email \in EmailsInQueue \Rightarrow \Diamond\Box(email \in Abandoned \cup Range(RemoteOutbox)))$

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

* Modification History
* Last modified *Wed* May 10 14:00:01 *KST* 2023 by *hcs*
* Created *Fri Apr* 28 13:04:37 *KST* 2023 by *hcs*