This is the BirthdayBook specification, from Spivey [1]. We extend it slightly by adding an extra operation, *RemindOne*, that is non-deterministic.

$$[NAME, DATE]$$

The *BirthdayBook* schema defines the *state space* of the birthday book system.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{BirthdayBook}} \\ \hline
\mathit{known} : \mathbb{P}\, \mathit{NAME} \\
\mathit{birthday} : \mathit{NAME} \mathrel{+\!\!\!\!\rightarrow} \mathit{DATE} \\ \hline
\mathit{known} = \operatorname{dom} \mathit{birthday} \\ \hline
\end{array}
$$

This *InitBirthdayBook* specifies the initial state of the birthday book system. It does not say explicitly that *birthday′* is empty, but that is implicit, because its domain is empty.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{InitBirthdayBook}} \\ \hline
\mathit{BirthdayBook}\ ' \\ \hline
\mathit{known}' = \{\} \\ \hline
\end{array}
$$

Next we have several operation schemas to define the normal (non-error) behaviour of the system.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{AddBirthday}} \\ \hline
\Delta \mathit{BirthdayBook} \\
\mathit{name}? : \mathit{NAME} \\
\mathit{date}? : \mathit{DATE} \\ \hline
(\mathit{name}? \notin \mathit{known} \\
\mathit{birthday}' = \mathit{birthday} \cup \{\mathit{name}? \mapsto \mathit{date}?\}) \\ \hline
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{FindBirthday}} \\ \hline
\Xi \mathit{BirthdayBook} \\
\mathit{name}? : \mathit{NAME} \\
\mathit{date}! : \mathit{DATE} \\ \hline
(\mathit{name}? \in \mathit{known} \\
\mathit{date}! = \mathit{birthday}(\mathit{name}?)) \\ \hline
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Remind}} \\ \hline
\Xi \mathit{BirthdayBook} \\
\mathit{today}? : \mathit{DATE} \\
\mathit{cards}! : \mathbb{P}\, \mathit{NAME} \\ \hline
\mathit{cards}! = \{n : \mathit{known} \mid \mathit{birthday}(n) = \mathit{today}?\} \\ \hline
\end{array}
$$

This *RemindOne* schema does not appear in Spivey, but is included to show how non-deterministic schemas can be animated. It reminds us of just one person who has a birthday on the given day.

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{RemindOne}} \\
\Xi \mathit{BirthdayBook} \\
\mathit{today}? : \mathit{DATE} \\
\mathit{card}! : \mathit{NAME} \\
\hline
(\mathit{card}! \in \mathit{known} \\
\mathit{birthday}\ \mathit{card}! = \mathit{today}?) \\
\hline
\end{array}$$

Now we strengthen the specification by adding error handling.

$\mathit{REPORT} ::= \mathit{ok} \mid \mathit{already\_known} \mid \mathit{not\_known}$

First we define auxiliary schemas that capture various success and error cases.

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Success}} \\
\mathit{result}! : \mathit{REPORT} \\
\hline
\mathit{result}! = \mathit{ok} \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{AlreadyKnown}} \\
\Xi \mathit{BirthdayBook} \\
\mathit{name}? : \mathit{NAME} \\
\mathit{result}! : \mathit{REPORT} \\
\hline
(\mathit{name}? \in \mathit{known} \\
\mathit{result}! = \mathit{already\_known}) \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{NotKnown}} \\
\Xi \mathit{BirthdayBook} \\
\mathit{name}? : \mathit{NAME} \\
\mathit{result}! : \mathit{REPORT} \\
\hline
(\mathit{name}? \notin \mathit{known} \\
\mathit{result}! = \mathit{not\_known}) \\
\hline
\end{array}$$

Finally, we define robust versions of all the operations by specifying how errors are handled. For illustration purposes, we leave the *RemindOne* operation non-robust.

$\mathit{RAddBirthday} == (\mathit{AddBirthday} \land \mathit{Success}) \lor \mathit{AlreadyKnown}$

$\mathit{RFindBirthday} == (\mathit{FindBirthday} \land \mathit{Success}) \lor \mathit{NotKnown}$

$\mathit{RRemind} == \mathit{Remind} \land \mathit{Success}$

References

[1] J. Michael Spivey. *The Z Notation: A Reference Manual.* International Series in Computer Science. Prentice-Hall International (UK) Ltd, second edition, 1992.