Homework  V  Sample  Solution

There are numerous ways to solve this problem, so your solution might be quite different than this one.

The basic types are unchanged so the first difference is the state space where an additional variable is incorporated to identify the listed phones, including an invariant condition pertaining to it.

PhoneDB

\[\text{members: } P \text{ Person}\]
\[\text{listed: } P \text{ Phone}\]
\[\text{telephones: Person } \Box \text{ Phone}\]

\[\text{dom telephones } \Box \text{ members}\]
\[\text{listed } \Box \text{ ran telephones}\]

It might be desirable to also change the initial state schema. However, this is actually not required -- since telephones = \(\emptyset\) and listed \(\Box\) ran telephones, it can be inferred that listed = \(\emptyset\).

Hence we proceed to the specifications of the new operations.

\[\text{UnList} \quad \text{__________________________}\]
\[\Box \text{PhoneDB}\]
\[\text{people?: } P \text{ Person}\]
\[\text{number?: Phone}\]

\[\text{number? } \Box \text{ listed}\]
\[\text{people?} = \text{telephones}\Box\{\text{number?}\}\]
\[\text{listed'} = \text{listed}\setminus\{\text{number?}\}\]
\[\text{telephones'} = \text{telephones}\]
\[\text{members'} = \text{members}\]
Since there are obviously exceptional conditions for the UnList operation, we add a schema to treat them.

\[
\text{UnListFailure} \quad \text{--------------------------}
\]
\[
\begin{align*}
\left[ & \Box \text{PhoneDB} \\
& \left[ \begin{array}{l}
\text{people}?: \mathcal{P} \text{Person} \\
\text{number}?: \text{Phone} \\
\text{rep}!: \text{Report} \\
\end{array} \right] \\
& \text{number}? \not\in \text{listed} \lor \text{people} \neq \text{telephones} \sim \{\text{number}?\} \\
& \text{rep}! = \text{UnList} \_\text{Error} \\
\end{align*}
\]

The completed operation is then defined by a schema expression in the usual way.

\[
\text{DoUnList} \overset{=}\leftarrow \text{UnList} \land \text{Success} \lor \text{UnList} \_\text{Failure}
\]

Now we go on to the second new operation.

\[
\text{ReList} \quad \text{--------------------------}
\]
\[
\begin{align*}
\left[ & \Box \text{PhoneDB} \\
& \left[ \begin{array}{l}
\text{people}?: \mathcal{P} \text{Person} \\
\text{number}?: \text{Phone} \\
\end{array} \right] \\
& \text{number}? \not\in \text{ran} \text{telephones} \land \text{number}? \not\in \text{listed} \\
& \text{people}? = \text{telephones} \sim \{\text{number}?\} \\
& \text{listed}' = \text{listed} \setminus \{\text{number}?\} \\
& \text{telephones}' = \text{telephones} \\
& \text{members}' = \text{members} \\
\end{align*}
\]

Again we need an exceptions schema to put together the complete operation specification.

\[
\text{ReListFailure} \quad \text{--------------------------}
\]
\[
\begin{align*}
\left[ & \Box \text{PhoneDB} \\
& \left[ \begin{array}{l}
\text{people}?: \mathcal{P} \text{Person} \\
\text{number}?: \text{Phone} \\
\text{rep}!: \text{Report} \\
\end{array} \right] \\
& \text{number}? \not\in \text{listed} \lor \text{number}? \not\in \text{ran} \text{telephones} \lor \text{people} \neq \text{telephones} \sim \{\text{number}?\} \\
& \text{rep}! = \text{UnList} \_\text{Error} \\
\end{align*}
\]

\[
\text{DoReList} \overset{=}\leftarrow \text{ReList} \land \text{Success} \lor \text{ReList} \_\text{Failure}
\]
In addition, we need to consider changes that are necessitated to other operation schemas to reflect and support this new feature. There turn out to be several that are warranted.

First of all, the AddEntry operation must be updated for the new state space.

```
AddEntry  ________________________________
PhoneDB
name?: Person
newnumber?: Phone

name?·members
name? → newnumber? · telephones
telephones' = telephones · {name? → newnumber?}
members' = members
listed' = listed · {name? → newnumber?}
```

This leaves the FindPhones and FindNames operations to be modified to honor refusing to reveal information about unlisted phones. First, the normal FindPhones operation makes sure all phones are listed.

```
FindPhones ________________
PhoneDB
name?: Person
numbers!: P Phone

name? · dom telephones
telephones( {name?} ) · listed
numbers! = telephones( {name?} )
```

Since this introduces an additional exception (i.e., the presence of an unlisted phone), we add a schema to treat it, plus revise the schema expression for the completed operation. This approach reports the listed numbers, and indicates the existence of an unlisted number.

```
UnListedPhone ________________
PhoneDB
name?: Person
numbers!: P Phone
rep!: Report

name? · dom telephones
telephones( {name?} ) \ listed ≠ {}
numbers! = listed · telephones( {name?} )
rep! = Unlisted_Phone
```
Lastly, we have similar changes for the FindNames operation.

\[
\text{FindNames} \quad \text{(PhoneDB)}
\]

\[
\begin{align*}
\text{name!}: & \quad P \text{ Person} \\
\text{number?}: & \quad \text{Phone} \\
\text{number? \ listed} \\
\text{names!} & = \text{telephones~\{-\{\text{number}\}\}}
\end{align*}
\]

An exceptional case is added so that for a known but unlisted number, no names are revealed.

\[
\text{UnlistedNumber} \quad \text{(PhoneDB)}
\]

\[
\begin{align*}
\text{number?}: & \quad \text{Phone} \\
\text{rep!}: & \quad \text{Report} \\
\text{number? \ ran telephones \ listed} \\
\text{rep!} & = \text{‘Unlisted Entry’}
\end{align*}
\]

\[
\text{DoFindNames} = \equiv \text{FindNames} \land \text{Success} \\
\lor \text{UnknownNumber} \lor \text{UnlistedNumber}
\]

Of course, the new error messages must also be added to the Report type.

The Zans version of this specification is in the class directory.