CMU 15-671 | Models of Software Systems | Fall 1997

Telephone Net

Garlan & Wing | Handout 8 | 6 October 1997

Preliminaries

We begin by adding a new operator to the Mathematical Tool-kit. *disj* is a predicate which is true of a sets of sets, S, if and only if no two distinct sets in S intersect.

$$\begin{array}{|l}
[X] \\
\hline
disj_ : \mathbb{P}(\mathbb{P}(\mathbb{P}\,X)) \\
\hline
\forall S : \mathbb{P}(\mathbb{P}\,X) \bullet \\
\quad disj\ S \Leftrightarrow (\forall i, j : S \bullet i \neq j \Rightarrow i \cap j =) \\
\hline
\end{array}$$

Basic Telephone Net

We take the type *PHONE* to be a given, or primitive type. A phone connection is simply a set of phones, and we define *CON* to represent the type of all possible phone connections.

$[PHONE]$

$CON == \mathbb{P}\,PHONE$

We now define the schema *TN* to model a telephone net. It consists of a set of connections, *cons*, and a set of requests, *reqs*. We require that all connections are also requests and that no phone is simultaneously in two distinct connections.

$$\begin{array}{|l}
TN \\
\hline
reqs, cons : \mathbb{P}\,CON \\
\hline
cons \subseteq reqs \\
disj\ cons \\
\hline
\end{array}$$

Efficient Telephone Net

An *efficient* telephone net is a telephone net for which it is not possible to connect an unsatisfied request without violating the telephone net invariant.

$$\begin{array}{|l}
EfficientTN \\
\hline
TN \\
\hline
\neg\,(\exists r : reqs \setminus cons \bullet disj\,(cons \cup \{r\})) \\
\hline
\end{array}$$

This schema expresses a property of maximality about the connections: it is not possible to enlarge the set of connections.

Operations

All operations that we plan to define for a telephone net will share certain properties. We capture these once and for all in the schema *Event*, which states that an operation will not disconnect any connections provided they are still requested after the operation.

$$\begin{array}{|l}\hline \textit{Event} \\ \Delta \textit{EfficientTN} \\ \hline \forall\, c : CON \bullet c \in (cons \cap reqs') \Rightarrow c \in cons' \\ \hline \end{array}$$

The operation *Call* adds a request to the set of requests of a telephone net.

$$\begin{array}{|l}\hline \textit{Call} \\ \textit{Event} \\ ph?, dialed? : PHONE \\ \hline reqs' = reqs \cup \{\{ph?, dialed?\}\} \\ \hline \end{array}$$

Given the maximality constraint on TN it is possible to infer from this operation that if neither phone is already connected, the request will be connected. (Exercise for the reader.)

The next operation, *Hangup*, terminates a connnection (if any) involving the phone $ph?$.

$$\begin{array}{|l}\hline \textit{Hangup} \\ \textit{Event} \\ ph? : PHONE \\ \hline reqs' = reqs \setminus \{c : cons \mid ph? \in c\} \\ \hline \end{array}$$

Finally, we define an operation *Busy*, which tells whether a phone is involved in a connection.

$$YesOrNo ::= YES \mid NO$$

$$\begin{array}{|l}\hline \textit{Busy} \\ \textit{Event} \\ ph? : PHONE \\ busy! : YesOrNo \\ \hline reqs' = reqs \\ busy! = YES \Leftrightarrow ph? \in \bigcup cons \\ \hline \end{array}$$

It is possible to show that *cons* does not change as a result of this operation. (Exercise for the reader.)