Telephones

Abstract
This is a small toy example which describes how the public telephone system – as it is conceived by a user (and not by a telephone technician). We ignore time-outs and special services such as conference calls etc.

The example is taken from Sect. 3.2 of Vol. 1 of the CPN book.

Developed and Maintained by:
Kurt Jensen, Aarhus University, Denmark (kjensen@daimi.au.dk).

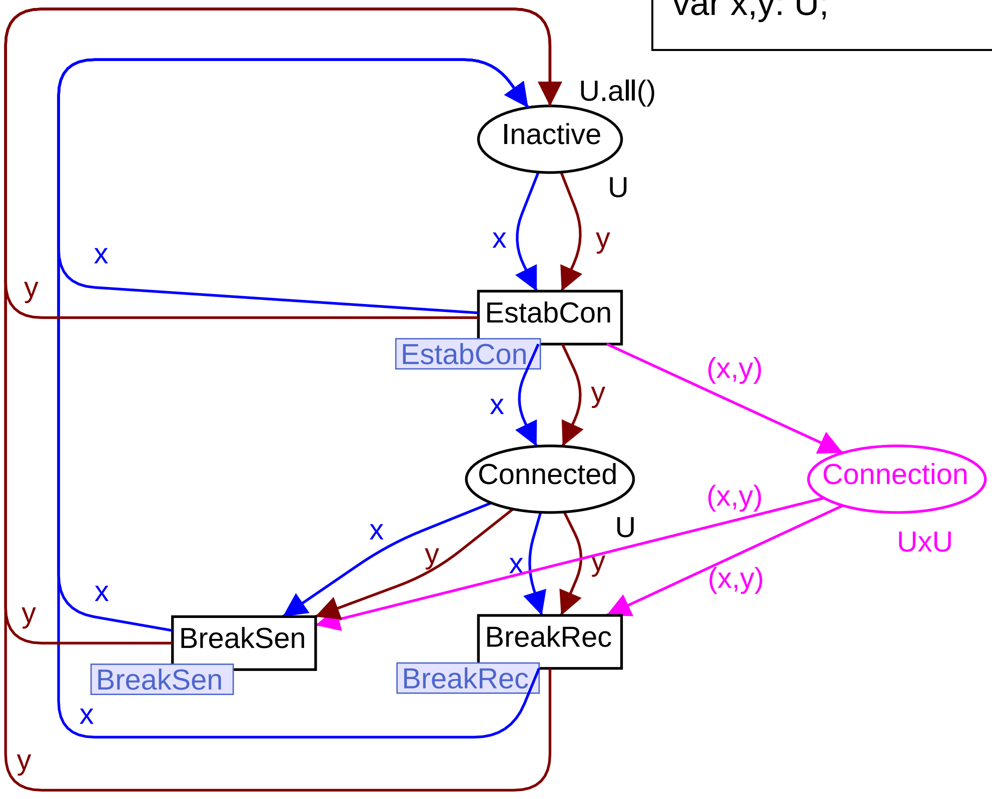
CPN Model

The telephone system has a single page *Phone*. This page has three transitions which all are substitution transitions. The substitution transitions are drawn with additional line thickness. However, we have hidden the hierarchy inscriptions because, in this particular system, they would only show that each substitution transition has a subpage with the same name, and that each socket node is assigned to a port node with the same name. From *Phone* we can see that the activities of the phone system are divided into three parts: the *Establishment of a Connection*, the *Breaking of a connection by the Sender*, and the *Breaking of a connection by the Recipient*.

The telephones are represented by U-tokens which at *Phone* only can be in two different states, *Inactive* and *Connected*. The telephone exchange (i.e., the electronics of the telephone switch) is represented by a single place, *Connection*. This place has a number of U×U–tokens, and each of them represents an established connection. The first element in a U×U–token identifies the **sender** (i.e., the phone from which the call was made), while the second element identifies the **recipient** (i.e., the phone which was called). It is important not to confuse the recipient with the receiver, which is the part of the telephone which you put to your ear. The receiver can be lifted and replaced, and we shall be a bit sloppy and say that the receiver is lifted or replaced by the sender/recipient – where we actually mean that a *person* at the sender/recipient performs the lifting/replacement operation. Analogously, we shall, e.g., say that

Phone

```
color U = index u with 1..10;
color UxU = product U * U;
var x,y: U;
```



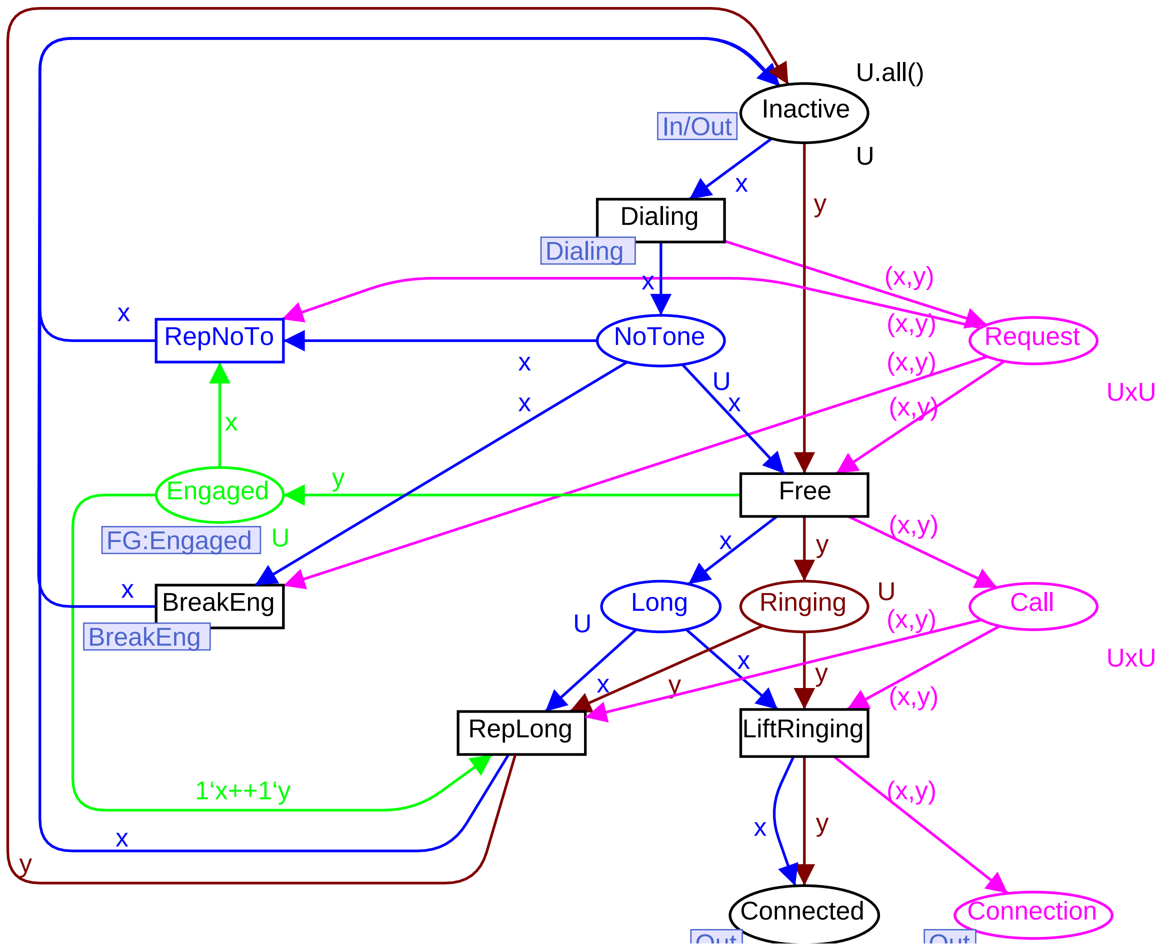
the sender dials a number.

Next, let us take a closer look at the *Establishment of a Connection*, i.e., the page *EstabCon*. This page has two substitution transitions, describing how the *Dialing of a number* and the *Breaking of a connection due to Engaged recipient* are performed. The telephones can now be in four additional states:

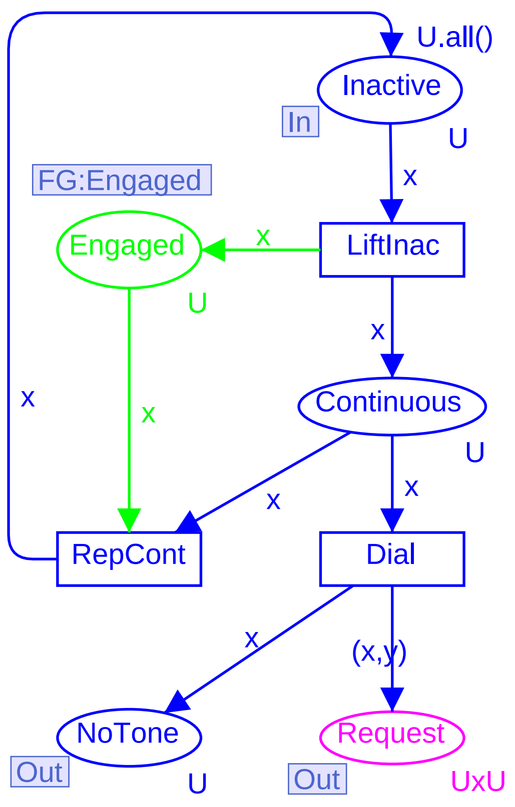
- *Engaged*, which intuitively is the opposite of *Inactive*. Notice that each transition which adds (removes) a token at *Inactive*, removes (adds) an identical token at *Engaged*. This property is not true for the substitution transitions, but this does not matter, because the arc expressions of the substitution transitions do not influence the behaviour of the hierarchical CP-net. We shall return to the arc expressions of the substitution transitions later in this section.

  The property above, and the initialization expressions of *Inactive* and *Engaged*, tell us that each telephone always has a token either at *Inactive* or at *Engaged*. Notice that *Engaged* is a member of a fusion set. This fusion set is called *FG:Engaged*, and it has five different places as members, which all are called *Engaged* and belong to five different pages.
- *NoTone*, which represents the state where no tone is heard at the sender in the few seconds where the system checks whether the recipient is *Inactive* or *Engaged*.

EstabCon



Dialing



- *Long*, which represents the state where a beep tone with long intervals is heard at the sender – indicating that the recipient is *Inactive*.
- *Ringing*, which represents the state where the recipient is ringing, while the sender is in state *Long*.

EstabCon has two additional places for the telephone exchange:

- *Request*, which represents demands for connections which have just been made, but not yet been established.
- *Call*, which represents demands for connections – where it has been checked that the recipient is *Inactive* but where the connection has not yet been established, because the recipient has yet to lift the receiver.

EstabCon has four transitions in addition to the two substitution transitions:

- *Free*, which models that the telephone exchange observes the recipient being *Inactive*. The transition brings the sender from *NoTone* to *Long* and the recipient from *Inactive* to *Ringing*.
- *LiftRinging*, which models that a *Call* is answered by lifting the receiver at the recipient.
- *RepNoTone*, which models the sender giving up and replacing the receiver, while in state *NoTone*.

- *RepLong*, which is analogous to RepNoTone, except that the sender is in state *Long*.

Next, let us consider how the *Dialing of a number* is performed, i.e., the page *Dialing*. This page has only one additional state for the telephones:

- *Continuous*, which represents the state where a continuous tone is heard – indicating that the sender now may dial a number.

Dialing has no additional places for the telephone exchange, and it has three transitions:

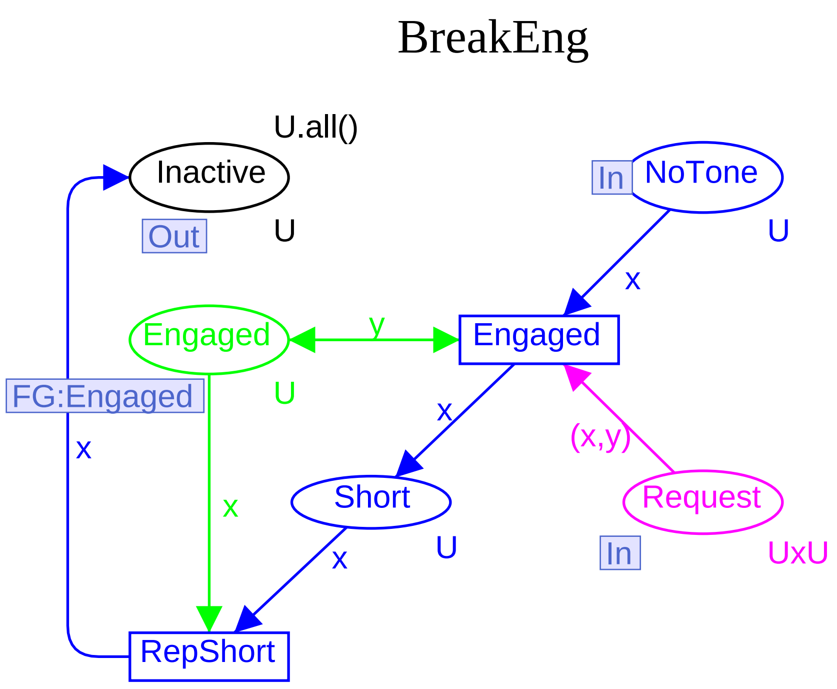
- *LiftInac*, which models that a sender lifts the receiver, while the corresponding phone is *Inactive*.
- *Dial*, which models that a number is dialled at the sender. Notice that the variable y only appears at the output arc expressions of *Dial*, and this means that it does not influence the enabling. Intuitively, this means that the sender can dial any number he wants – including his own.
- *RepCont*, which is analogous to the other replacement transitions, except that the sender is in state *Continuous*.

Next, let us consider how the *Breaking of a connection due to Engaged recipient* is performed, i.e., the page *BreakEng*. This page has only one additional state for the telephones:

- *Short*, which represents the state where a beep tone with short intervals is heard at the sender – indicating that the recipient is *Engaged*.

BreakEng has no additional places for the telephone exchange, and it has two transitions:

- *Engaged*, which models that the telephone exchange observes the recipient being *Engaged*. The transition brings the sender from *NoTone* to *Short* while the state of the recipient is unaltered.
- *RepShort*, which is analogous to the other replacement transitions, except that the sender is in state *Short*.



Next, let us consider how the *Breaking of a connection by the Sender* is performed, i.e., the page *BreakSen*. This page has only one additional state for the telephones:

- *Disconnected*, which represents the state where the recipient has become disconnected because the sender has broken the connection by replacing the receiver.

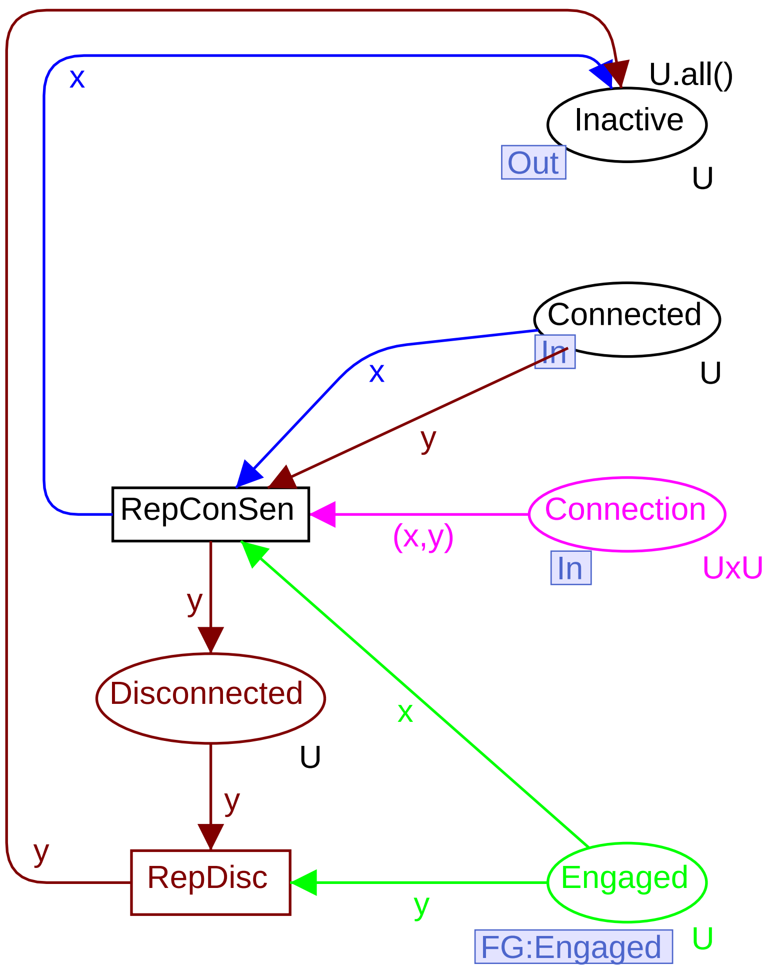
BreakSen has no additional places for the telephone exchange, and it has two transitions:

- *RepConSen*, which models that the sender breaks the connection by replacing the receiver. Notice that the effect of this transition is *not* totally analogous to the other replacement transitions, because it is only the sender that returns to *Inactive*, while the recipient becomes *Disconnected*.
- *RepDisc*, which is analogous to the other replacement transitions, except that the recipient is in state *Disconnected*.

Finally, let us consider how the *Breaking of a connection by the Recipient* is performed, i.e., the page *BreakRec*. This page has only one additional state for the telephones:

- *Replaced*, which represents the state where the recipient has replaced the receiver. Notice that this action – at least in the Danish telephone system – does not terminate the connection, and this means that the conversation can be continued, if the recipient again lifts the receiver (for more information, see the description of *RepConRec* and *LiftRepl* below).

BreakSen

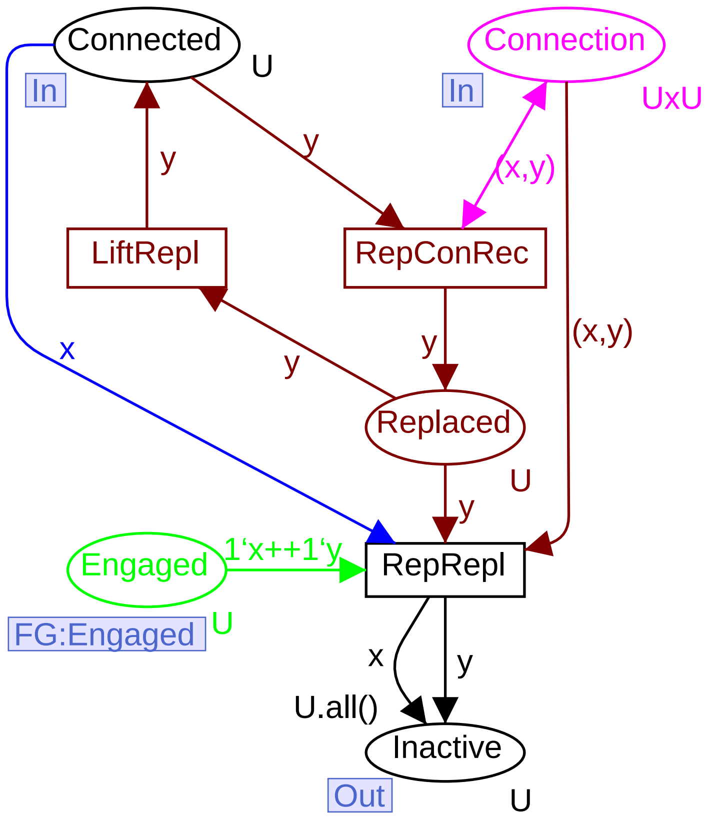


BreakRec has no additional places for the telephone exchange, and it has three transitions:

- *RepConRec*, which models that the recipient replaces the receiver. Notice that the effect of this transition is *not* analogous to the other replacement transitions, because neither the sender nor the recipient returns to *Inactive*. Instead the sender remains *Connected*, while the recipient becomes *Replaced*.
- *LiftRepl*, which represents the fact that the recipient lifts the receiver, while the recipient is in state *Replaced*.
- *RepRepl*, which is analogous to the other replacement transitions, except that the sender is in state *Connected* while the recipient is in state *Replaced*.

Now let us take a closer look at the arc expressions which surround the substitution transitions. These arc expressions do not influence the behaviour of the hierarchical CP-net, because the substitution transitions and their surrounding arcs are replaced by the subpages, and this means that it does not make sense to talk about an enabled or an occurring substitution transition. Thus we can omit the arc expressions surrounding the substitution transitions.

BreakRec



However, it is also possible to specify these arc expressions in a less trivial way – and they then act rather like comments, giving the reader of the hierarchical CP-net an idea about the behaviour of the subpages.

In the telephone example the arc expressions of the substitution transitions are used in two slightly different ways. The arc expressions of *BreakSen* give a rather precise description of the behaviour of the subpage *BreakSen*. The only differences are that the arc expressions (of course) do not show that the activity is executed in two subsequent steps, and that the arc expressions do not show the updating of *Engaged* (which does not exist at the abstraction level of *Phone)*. The arc expressions of *EstabCon* are used in a different way. These arc expressions only describe the standard case, i.e., the case in which a *Connection* is established, and they ignore the possibilities where the recipient is *Engaged* or the sender replaces the receiver. It would not have been particularly difficult to include these additional possibilities into the arc expressions. However, the modeller has chosen, at the abstraction level of Phone, to concentrate on the standard case.

It may be argued that it is necessary to force the modeller to have a one-to-one correspondence between the arc expressions of a substitution transition and the behaviour of the corresponding subpage – because this will make it easier to develop methods allowing modular analysis. We think it is very important to develop such incremental analysis methods, built upon strict behavioural equivalence between the different levels of description. However, as indicated above, we do not think that strict behavioural equivalence is the only interesting relationship between the different abstraction levels. As an example, we may also want (for the hierarchical telephone system) to model the possibility of time-outs and the existence of special services such as conference calls, etc. This can be done by adding a number of transitions to the existing pages, but it can also be done by turning the existing transitions into substitution transitions. Then each subpage will model the effects of the time-outs and special services – as well as the standard behaviour of the corresponding action. This will mean that the pages shown above describe the standard behaviour of the telephone system, while the additional complexity introduced by time-outs and the special services are confined to the new pages. There will be a consistent relationship between the two levels of description – but this relationship will not be a strict behavioural equivalence.

The telephone system was presented in a **top-down** manner, but this does not necessarily mean that it was constructed in this way. It could just as well have been constructed **bottom-up** or (more likely) by mixing the two strategies.