

## Appendix 7

### Telephone animation and testing

This Appendix contains examples of possible behaviours that the LSD specification in chapter 6 of a telephone system can produce when simulated. Appendix 7.1 contains the results of executing the adm program in §6.3. Appendices 7.2 to 7.9 contain the output from execution of other valid transformations of the LSD specification.

The purpose of this appendix is demonstrate how one specification can be transformed into a family of definitive programs by using different parameters for the transformation process.

#### Appendix 7.1 The transformed telephone program

In this appendix we give a listing of the complete program of §6.3.

**entity** user(\_U,\_S,\_D) {

**definition**

```
number_to_dial[_S] = _D,      tone[_S,1] = |tone[_S]],
ringing[_S,1] = |ringing[_S]], onhook[_S,1] = |onhook[_S]],
dialled_number[_S,1] = |dialled_number[_S]], level[_U,1] = 0,
slow[_U,1] = 0, only_one[_U,1] = 1, selected[_U,1] = 0
```

**action**

```
even && ringing[_S,1] print (_S," is ringing") -> ,
even && (slow[_U,1] > 0) -> slow[_U,1] = |slow[_U,1]] - 1,
even && !onhook[_S,1] && (level[_U,1] == 0) && (only_one[_U,1] <= 2) &&
    (selected[_U,1] == 0) && (slow[_U,1] == 0)
    -> selected[_U,1] = 1 ; level[_U,1] = 1,
even && (selected[_U,1] == 1) && (level[_U,1] == 1) && (rand(3) == 1)
    print ("replacing the receiver of ",_S) -> onhook[_S]=true ; level[_U,1] = 2,
even && (selected[_U,1] == 1) && (level[_U,1] == 2)
    -> dialled_number[_S]=0 ; level[_U,1] = 0 ; selected[_U,1] = 0,
even && !onhook[_S,1] && (tone[_S,1] == 1) && (level[_U,1] == 0) && (only_one[_U,1] >= 3) &&
(only_one[_U,1] <= 17) && (selected[_U,1] == 0) && (slow[_U,1] == 0)
    -> selected[_U,1] = 2 ; level[_U,1] = 1,
even && (selected[_U,1] == 2) && (level[_U,1] == 1) && (rand(6) == 1)
    print ("dialling the number ",_D," on telephone ",_S)
    -> dialled_number[_S]= number_to_dial[_S] ;
    level[_U,1] = 0 ; slow[_U,1] = 10 ; selected[_U,1] = 0,
even && !onhook[_S,1] && (tone[_S,1] == 4) && (level[_U,1] == 0) &&
(only_one[_U,1] >= 18) && (only_one[_U,1] <= 27)
    && (selected[_U,1] == 0) && (slow[_U,1] == 0)
    -> selected[_U,1] = 3 ; level[_U,1] = 1,
even && (selected[_U,1] == 3) && (level[_U,1] == 1) && (rand(6) == 1)
    print ("speaking into ",_S) -> level[_U,1] = 0 ; selected[_U,1] = 0,
even && onhook[_S,1] && !ringing[_S,1] && (level[_U,1] == 0) && (only_one[_U,1] >= 28) &&
(only_one[_U,1] <= 29) && (selected[_U,1] == 0) && (slow[_U,1] == 0)
    -> selected[_U,1] = 4 ; level[_U,1] = 1,
even && (selected[_U,1] == 4) && (level[_U,1] == 1) && (rand(3) == 1)
    print ("lifting up receiver of ",_S," to make call")
    -> onhook[_S] = false; level[_U,1] = 2,
even && (selected[_U,1] == 4) && (level[_U,1] == 2)
    -> dial(_S,_D) ; level[_U,1] = 0 ; selected[_U,1] = 0,
even && onhook[_S,1] && ringing[_S,1] && (level[_U,1] == 0) && (only_one[_U,1] >= 30)
    && (only_one[_U,1] <= 39) && (selected[_U,1] == 0) && (slow[_U,1] == 0)
    -> selected[_U,1] = 5 ; level[_U,1] = 1,
```

```

even && (selected[_U,1] == 5) && (level[_U,1] == 1) && (rand(3) == 1)
    print ("lifting up receiver of ",_S," to answer")
    -> onhook[_S] = false; level[_U,1] = 2,
even && (selected[_U,1] == 5) && (level[_U,1] == 2) && (rand(6) == 1)
    print ("hello, ",_S,"")
    -> level[_U,1] = 0 ; selected[_U,1] = 0,
!even -> tone[_S,1] = |tone[_S]|,
!even -> ringing[_S,1] = |ringing[_S]|,
!even -> onhook[_S,1] = |onhook[_S]|,
!even -> dialled_number[_S,1] = |dialled_number[_S]|,
!even && (selected[_U,1] == 0) -> only_one[_U,1] = |rand(50)|
}

```

**entity** telephone(\_S,\_D) {

**definition**

```

onhook[_S] = true, dialled_number[_S] = 0,
connected[_S,_D,2] = |connected[_S,_D]|,
connecting[_S,_D] = false, engaged[_S,_D,2] = |engaged[_S,_D]|,
dialling[_S] = false,
tone[_S] = if dialling[_S] then 1 else
    if (connecting[_S,_D] && engaged[_S,_D]) then 2 else
    if (connected[_S,_D] && onhook[_D]) then 3 else
    if ((connected[_S,_D] || connected[_D,_S]) && !onhook[_D]) then 4 else 5

```

**action**

```

!even -> connected[_S,_D,2]=|connected[_S,_D]|,
!even -> engaged[_S,_D,2] = |engaged[_S,_D]|
}

```

**entity** dial(\_S,\_D) {

**definition**

```

dialled_number[_S,3] = |dialled_number[_S]|,
Tdial[_S,3] = |Tdial|, time[_S,3] = |time|,
onhook[_S,3] = |onhook[_S]|,
connecting[_S,_D,3] = |connecting[_S,_D]|,
tstart[_S] = |time|, valid[_S] = (dialled_number[_S] != 0),
dialling[_S,3] = !|onhook[_S,3] && ((time[_S,3] - tstart[_S]) < Tdial[_S,3])|,
init_flag[_S,3] = true, level[_S,3] = 0, slow[_S,3] = 0

```

**action**

```

init_flag[_S,3] -> dialling[_S] = !|onhook[_S] && ((time - tstart[_S]) < Tdial) ;
    tstart[_S] = |time| ; init_flag[_S,3] = false,
even && (slow[_S,3] > 0) -> slow[_S,3] = |slow[_S,3]| - 1,
even && (slow[_S,3] == 0) && dialling[_S,3] && valid[_S] && (level[_S,3] == 0)
    -> level[_S,3] = 1,
even && (level[_S,3] == 1)
    print ("starting connection from ",_S," to ",_D)
    -> connect(_S,_D) ; level[_S,3] = 0 ; slow[_S,3] = 20,
!even && !init_flag[_S,3] && !(dialling[_S,3] && !connecting[_S,_D,3])
    print ("terminating dialler from ",_S," to ",_D)
    -> dialling[_S] = false ; delete dial(_S,_D),
!even && dialling[_S,3] && !connecting[_S,_D,3] && (rand(2) == 1) && !init_flag[_S,3]
    -> dialled_number[_S,3] = |dialled_number[_S]|,
!even && dialling[_S,3] && !connecting[_S,_D,3] && (rand(10) == 1) && !init_flag[_S,3]
    -> Tdial[_S,3] = |Tdial|,
!even && dialling[_S,3] && !connecting[_S,_D,3] && !init_flag[_S,3]
    -> time[_S,3] = |time|,
!even && dialling[_S,3] && !connecting[_S,_D,3] && !init_flag[_S,3]
    -> onhook[_S,3] = |onhook[_S]|,
!even && dialling[_S,3] && !connecting[_S,_D,3] && !init_flag[_S,3]
    -> connecting[_S,_D,3] = |connecting[_S,_D]|,
!even && dialling[_S,3] && !connecting[_S,_D,3] && !init_flag[_S,3]
    -> dialling[_S,3] = |dialling[_S]|
}

```

**entity** connect(\_S,\_D) {

## definition

```
onhook[_S,_S,4] = |onhook[_S]|, onhook[_S,_D,4] = |onhook[_D]|,
ringing[_D,4] = |ringing[_D]|, Tcall[_S,4] = |Tcall|,
time[_S,4] = |time|, tcall[_S] = |time|,
connected[_S,_D,4] = false, answered[_S,_D,4] = false, connecting[_S,_D,4] = !
connected[_S,_D,4],
engaged[_S,_D,4] = |engaged[_S,_D]|, init_flag[_S,4] = true,
level[_S,4] = 0, only_one[_S,_D,4] = 3,
selected[_S,_D,4] = 0
```

## action

```
init_flag[_S,4] -> answered[_D] = false ;
tcall[_S] = |time| ;
connected[_S,_D] = false ;
connecting[_S,_D] = !connected[_S,_D] ; init_flag[_S,4] = false,
even && !engaged[_S,_D,4] && !connected[_S,_D,4] && (only_one[_S,_D,4] == 1) &&
(selected[_S,_D,4] == 0) && (rand(3) == 1) && (level[_S,4] == 0)
-> level[_S,4] = 1 ; selected[_S,_D,4] = 1,
even && (level[_S,4] == 1) && (selected[_S,_D,4] == 1)
print ("connection made from ",_S," to ",_D)
-> connected[_S,_D] = true; level[_S,4] = 0 ; selected[_S,_D,4] = 0,
even && !onhook[_S,_D,4] && !answered[_S,_D,4] && connected[_S,_D,4]
&& (only_one[_S,_D,4] == 2) && (selected[_S,_D,4] == 0) && (level[_S,4] == 0)
-> level[_S,4] = 1 ; selected[_S,_D,4] = 2,
even && (level[_S,4] == 1) && (selected[_S,_D,4] == 2)
print ("registered picking up of receiver at ",_D)
-> answered[_D] = true ; level[_S,4] = 0 ; selected[_S,_D,4] = 0,
even && engaged[_S,_D,4] && (level[_S,4] == 0)
&& (only_one[_S,_D,4] == 3) && (selected[_S,_D,4] == 0)
print ("no connection possible from ",_S," to ",_D," because line is engaged")
-> level[_S,4] = 1 ; selected[_S,_D,4] = 3,
even && (level[_S,4] == 1) && (selected[_S,_D,4] == 3)
-> connecting[_S,_D] = false; connected[_S,_D] = false; delete connect(_S,_D),
!even && !init_flag[_S,4] && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
print ("connection from ",_S," to ",_D," broken")
-> connected[_S,_D] = false; connecting[_S,_D] = false; delete connect(_S,_D),
!even && (rand(2) == 1) && !init_flag[_S,4] && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
-> onhook[_S,_S,4] = |onhook[_S]|,
!even && (rand(2) == 1) && !init_flag[_S,4] && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
-> onhook[_S,_D,4] = |onhook[_D]|,
!even && !init_flag[_S,4] && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
-> ringing[_D,4] = |ringing[_D]|,
!even && !init_flag[_S,4] && (rand(10) == 1) && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
-> Tcall[_S,4] = |Tcall|,
!even && !init_flag[_S,4] && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
|| ringing[_D]
|| (connected[_S,_D] && (!answered[_D] || !onhook[_D]))))
-> time[_S,4] = |time|,
!even && !init_flag[_S,4] && (rand(2) == 1) && (!onhook[_S]
&& ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
```

```

        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        -> engaged[_S,_D,4] = |engaged[_S,_D]|,
!even && !init_flag[_S,4] && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        -> connected[_S,_D,4] = |connected[_S,_D]|,
!even && !init_flag[_S,4] && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        -> answered[_S,_D,4] = |answered[_D]|,
!even && !init_flag[_S,4] && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        -> connecting[_S,_D,4] = |connecting[_S,_D]|,
!even && (selected[_S,_D,4] == 0) && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        && !engaged[_S,_D,4] && !connected[_S,_D,4]
        -> only_one[_S,_D,4] = 1,
!even && (selected[_S,_D,4] == 0) && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        && !onhook[_S,_D,4] && !answered[_S,_D,4] && connected[_S,_D,4]
        -> only_one[_S,_D,4] = 2,
!even && (selected[_S,_D,4] == 0) && (!onhook[_S]
    && ( (connecting[_S,_D] && ((time-tcall[_S])<Tcall))
        || ringing[_D]
        || (connected[_S,_D] && (!answered[_D] || !onhook[_D])))
        && engaged[_S,_D,4]
        -> only_one[_S,_D,4] = 3
}

```

**entity** exchange() {

**definition**

```

    onhook[6489,5] = |onhook[6489]|, onhook[7124,5] = |onhook[7124]|,
    connected[6489,7124] = false, connected[7124,6489] = false,    connecting[6489,7124,5] = |
connecting[6489,7124]|,
    connecting[7124,6489,5] = |connecting[7124,6489]|,
    answered[6489] = false, answered[7124] = false,
    ringing[7124] = connected[6489,7124] && onhook[7124] && !answered[7124],
    ringing[6489] = connected[7124,6489] && onhook[6489] && !answered[6489],
    engaged[6489,7124] = connecting[6489,7124,5] && (ringing[7124] || !onhook[7124]),
    engaged[7124,6489] = connecting[7124,6489,5] && (ringing[6489] || !onhook[6489]),
    Tdial = 330, Tcall = 340

```

**action**

```

    !even && (rand(2) == 1) -> onhook[6489,5] = |onhook[6489]|,
    !even && (rand(2) == 1) -> onhook[7124,5] = |onhook[7124]|,
    !even -> connecting[6489,7124,5] = |connecting[6489,7124]|,
    !even -> connecting[7124,6489,5] = |connecting[7124,6489]|
}

```

**entity** environment() {

**definition**

```

    time = 0, level[6] = 0, even = 0

```

**action**

```

    even && (level[6] == 0) -> level[6] = 1,
    even && (level[6] == 1) -> time = |time| + 1 ; level[6] = 0,

```

```

    true -> even = !|even|
}

```

## Appendix 7.2 Execution of the program

The parameters used for the transformation which resulted in this program are such that it is unlikely that a user will either initiate or terminate a call. This is represented by the way that the ratios for the five user actions are:

```

2      replace receiver
15     dial number
10     speak when connected
2      start a call
10     answer phone.

```

These ratios are the ranges for the variable `only_one[_U,1]` in the `user()` entity which will result in the respective guards being the one chosen for evaluation in an execution cycle. It is therefore five times as likely that, once connected, the user will speak rather than terminate the call by replacing the receiver. For this reason we would expect to have reasonable length calls with this simulation.

For this appendix we show the results of execution using numbering of procedural action output. Notice the large number of execution cycles which occur between the termination of a call and the erroneous speaking by the user of telephone 7124. The relevant guard to explain this in the program of §7.3 is the ninth guard in the user entity, which has a conjunct

```
&& (rand(6) == 1)
```

This means that, once it has been decided to speak, on average only one of the next twelve (since the guard can only be true when even is true) execution cycles will allow speaking to occur.

The value for `n` used in the guarded command

```
!even && (selected[_U,1] == 0) -> only_one[_U,1] = |rand(n)|
```

influences how quickly actions are selected by the user. The higher the value for `n`, the more execution cycles will on average occur between actions by the user. This is because any value for `only_one[_U,1]` which is more than 39 will not allow the selection of any guard from the specification. The higher `n` is, the more chance there is that `only_one[_U,1]` will have a value higher than 39.

In the simulation it can be seen that the user sometimes decides to speak when the connection between the two telephones is in place, but only actually does the speaking when the connection has been broken. This indicates a problem with the specification, which does not represent the way there is implicit agreement in a conversation not to put the telephone down with no warning. The LSD specification could therefore be extended to include synchronisation between listening and speaking, and to contain a protocol for ending the conversation, such as the saying of "goodbye" at both ends of the connection.

```

Script started on Mon Aug 28 12:39:48 1989
emerald@mike am < program
am> compiling user(_U,_S,_D)
am> compiling telephone(_S,_D)
am> compiling dial(_S,_D)
am> compiling connect(_S,_D)    compile the various entities
am> compiling exchange()
am> compiling environment()
am> instantiating user(10,6489,7124)
am> instantiating user(11,7124,6489)
am> instantiating telephone(6489,7124)
am> instantiating telephone(7124,6489)    instantiate the various entities
am> instantiating exchange
am> instantiating environment
am> starting simulation

```

```

(40) lifting up receiver of 7124 to make call
(64) dialling the number 6489 on telephone 7124

```

(68) starting connection from 7124 to 6489  
(73) terminating dialler from 7124 to 6489  
(76) connection made from 7124 to 6489  
(78) 6489 is ringing  
(80) 6489 is ringing  
(82) 6489 is ringing  
(84) 6489 is ringing  
(86) 6489 is ringing  
(86) lifting up receiver of 6489 to answer  
(94) registered picking up of receiver at 6489  
(100) 'hello, 6489'  
(106) replacing the receiver of 7124  
(107) connection from 7124 to 6489 broken  
(112) replacing the receiver of 6489 both receivers are now on the hook

(154) lifting up receiver of 7124 to make call  
(176) dialling the number 6489 on telephone 7124  
(180) starting connection from 7124 to 6489  
(185) terminating dialler from 7124 to 6489  
(186) connection made from 7124 to 6489  
(188) 6489 is ringing  
(200) 6489 is ringing  
(202) 6489 is ringing  
(204) 6489 is ringing  
(206) 6489 is ringing  
(206) lifting up receiver of 6489 to answer  
(212) speaking into 7124  
(212) registered picking up of receiver at 6489  
(228) 'hello, 6489'  
(228) speaking into 7124 both people speak at the same time  
(242) speaking into 6489  
(242) speaking into 7124 the synchronisation in this execution cycle is coincidental  
(264) speaking into 7124  
(270) speaking into 6489  
(276) replacing the receiver of 6489  
(277) connection from 7124 to 6489 broken  
(300) lifting up receiver of 6489 to make call 7124 is still off the hook  
(304) speaking into 7124 the user of 7124 had decided to speak when  
it was still a valid option, i.e. when the connection  
between 6489 and 7124 was in use.

(320) replacing the receiver of 6489  
(323) terminating dialler from 6489 to 7124  
(324) replacing the receiver of 7124 both receivers are now on the hook  
(350) lifting up receiver of 7124 to make call  
(370) dialling the number 6489 on telephone 7124  
(374) starting connection from 7124 to 6489  
(379) terminating dialler from 7124 to 6489  
(392) connection made from 7124 to 6489  
(394) 6489 is ringing  
(396) 6489 is ringing  
(398) 6489 is ringing  
(400) 6489 is ringing  
(402) 6489 is ringing  
(402) lifting up receiver of 6489 to answer  
(404) 'hello, 6489'  
(412) speaking into 6489  
(412) registered picking up of receiver at 6489  
(416) speaking into 7124  
(430) speaking into 6489  
(438) replacing the receiver of 6489  
(439) connection from 7124 to 6489 broken  
(476) speaking into 7124 again the user of 7124 had decided to speak when  
it was still a valid option, i.e. when the connection



lifting up receiver of 6489 to make call  
replacing the receiver of 6489  
terminating dialler from 6489 to 7124        because 6489 is now on the hook  
lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
connection made from 7124 to 6489  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
lifting up receiver of 6489 to answer  
registered picking up of receiver at 6489  
replacing the receiver of 7124  
connection from 7124 to 6489 broken  
'hello, 6489'        decided to speak when the connection was still existent

script done on Mon Aug 28 12:32:16 1989

Appendix 7.4 Answering more often than initiating or terminating a call

The parameters used for the transformation which resulted in the program whose behaviour is shown in this appendix are such that a user is more likely to answer a call than initiate or terminate a call. This is represented by the way that the ratios for the five user actions are:

1        replace receiver  
10       dial number  
3        speak when connected  
1        start a call  
9        answer phone.

The variable `only_one[_U,1]` is assigned a random value between 1 and 40. This simulation shows how a line can be recognised as engaged, and how it takes time for the consequences of putting a telephone back on the hook when it is calling another telephone to become known.

Script started on Mon Aug 28 12:11:24 1989

emerald!mike am -s < program2

lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
connection made from 7124 to 6489  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
lifting up receiver of 6489 to answer  
'hello, 6489'  
registered picking up of receiver at 6489  
replacing the receiver of 6489  
connection from 7124 to 6489 broken  
speaking into 7124        decided to speak when the connection was still existent  
lifting up receiver of 6489 to make call  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124        7124 is off the hook  
no connection possible from 6489 to 7124 because line is engaged  
terminating dialler from 6489 to 7124  
replacing the receiver of 7124  
replacing the receiver of 6489        both receivers on the hook  
lifting up receiver of 6489 to make call  
dialling the number 7124 on telephone 6489



starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
connection made from 6489 to 7124  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
replacing the receiver of 6489  
7124 is ringing  
connection from 6489 to 7124 broken  
7124 is ringing  
lifting up receiver of 7124 to answer           propagation of change to  
'hello, 7124'       onhook[6489] not quick enough  
lifting up receiver of 6489 to make call  
replacing the receiver of 6489  
terminating dialler from 6489 to 7124  
lifting up receiver of 6489 to make call  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124  
no connection possible from 6489 to 7124 because line is engaged  
terminating dialler from 6489 to 7124  
replacing the receiver of 7124  
lifting up receiver of 7124 to make call  
replacing the receiver of 7124  
terminating dialler from 7124 to 6489  
lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
no connection possible from 7124 to 6489 because line is engaged  
terminating dialler from 7124 to 6489  
replacing the receiver of 7124  
replacing the receiver of 6489

script done on Mon Aug 28 12:14:16 1989  
Appendix 7.5 More calling than terminating of calls

The parameters used for the transformation which resulted in the program whose behaviour is shown in this appendix are such that a user is more likely to initiate a call than initiate or terminate a call. This is represented by the way that the ratios for the five user actions are:

2	replace receiver
10	dial number
20	speak when connected
5	start a call
20	answer phone.

The variable `only_one[_U,1]` is assigned a random value between 1 and 100. This simulation shows how an unwillingness to replace receivers can lead to lines often being engaged.

Script started on Mon Aug 28 12:34:29 1989  
emerald!mike am -s < program

lifting up receiver of 7124 to make call  
lifting up receiver of 6489 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
no connection possible from 7124 to 6489 because line is engaged  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
no connection possible from 6489 to 7124 because line is engaged  
replacing the receiver of 7124  
lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
no connection possible from 7124 to 6489 because line is engaged  
replacing the receiver of 7124  
replacing the receiver of 6489  
lifting up receiver of 7124 to make call  
lifting up receiver of 6489 to make call  
replacing the receiver of 7124  
terminating dialler from 7124 to 6489  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
connection made from 6489 to 7124  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
7124 is ringing  
lifting up receiver of 7124 to answer  
'hello, 7124'  
registered picking up of receiver at 7124  
speaking into 7124  
speaking into 6489  
speaking into 7124  
speaking into 6489  
speaking into 6489  
speaking into 7124  
speaking into 6489  
replacing the receiver of 7124  
connection from 6489 to 7124 broken  
lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
no connection possible from 7124 to 6489 because line is engaged  
replacing the receiver of 6489  
lifting up receiver of 6489 to make call  
replacing the receiver of 7124  
lifting up receiver of 7124 to make call  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
no connection possible from 6489 to 7124 because line is engaged  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
no connection possible from 7124 to 6489 because line is engaged  
replacing the receiver of 6489  
lifting up receiver of 6489 to make call  
dialling the number 7124 on telephone 6489

starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
no connection possible from 6489 to 7124 because line is engaged  
replacing the receiver of 6489  
lifting up receiver of 6489 to make call  
dialling the number 7124 on telephone 6489  
starting connection from 6489 to 7124  
terminating dialler from 6489 to 7124  
no connection possible from 6489 to 7124 because line is engaged  
replacing the receiver of 6489  
replacing the receiver of 7124

script done on Mon Aug 28 12:38:22 1989  
Appendix 7.6 Removal of the slowing down of the user

In the program of §6.3, the user is slowed down after dialling a number. This stops the user from participating in the next 20 execution cycles. The reason for this is that the tone is not immediately updated, so one potential behaviour described by the LSD specification is for the user to dial the same number twice. This is shown in the simulation in this appendix. The slowing down mechanism is removed from the definitive program, which results in the following behaviour:

Script started on Mon Aug 28 12:03:06 1989  
emerald!mike am -s < program  
lifting up receiver of 7124 to make call  
dialling the number 6489 on telephone 7124  
starting connection from 7124 to 6489  
terminating dialler from 7124 to 6489  
connection made from 7124 to 6489  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
6489 is ringing  
lifting up receiver of 6489 to answer  
dialling the number 6489 on telephone 7124 unintended behaviour

script done on Mon Aug 28 12:03:37 1989  
Appendix 7.7 Allowing the dialler to time out

In the program of §6.3 the value for Tdial is set to a very high number, viz 330. This allows nearly all simulations to execute without the dialler terminating due to dialling taking too long. The following simulation shows how changing the value for Tdial from 330 to 5 may cause the dialler to terminate before the number has been dialled.

Script started on Mon Aug 28 12:47:56 1989  
emerald!mike am -s < program6  
lifting up receiver of 7124 to make call  
terminating dialler from 7124 to 6489  
dialling the number 6489 on telephone 7124  
replacing the receiver of 7124

script done on Mon Aug 28 12:48:31 1989  
Appendix 7.8 Dialling quicker

If the dialler can time out after only 5 time units then the user must be capable of dialling more quickly than in the program of §6.3. To allow this we change the rand(6) in the ninth guard of the program of §6.3 into rand(2). This decreases the average time spent by the user in dialling, and results in the following simulation:

Script started on Mon Aug 28 12:49:24 1989  
emerald!mike am -s < program7

lifting up receiver of 7124 to make call  
 dialling the number 6489 on telephone 7124 quickly enough to avoid time out  
 starting connection from 7124 to 6489  
 terminating dialler from 7124 to 6489  
 connection made from 7124 to 6489  
 6489 is ringing  
 6489 is ringing  
 6489 is ringing  
 6489 is ringing  
 lifting up receiver of 6489 to answer  
 registered picking up of receiver at 6489  
 'hello, 6489' speaking into 7124  
 speaking into 6489  
 speaking into 7124  
 speaking into 6489  
 replacing the receiver of 7124  
 connection from 7124 to 6489 broken  
 replacing the receiver of 6489

script done on Mon Aug 28 12:50:48 1989

Similar issues with the connect() entity time out mechanisms arise when considering the time taken to make the connection.

#### Appendix 7.9 Interactive use of the telephone simulation

The definitive programs used in Appendices 7.1 to 7.8 have all executed non-interactively. In this appendix we show how the program of §6.3 can be executed interactively, and the additional control that this gives over execution. The various stores of the abstract definitive machine are listed to check that they have stored the program correctly, and then the interactive capability is used to select that the next action by user 10 is to put telephone 6489 back on the hook.

Script started on Mon Aug 28 18:46:04 1989

```

emerald!mike cat -u program5 - | am -i1      invoke am with iterations set to 1
am> compiling user(_U,_S,_D)
am> compiling telephone(_S,_D)
am> compiling dial(_S,_D)
am> compiling connect(_S,_D)
am> compiling exchange()
am> compiling environment()
am> instantiating user(10,6489,7124)
am> instantiating user(11,7124,6489)
am> instantiating telephone(6489,7124)
am> instantiating telephone(7124,6489)
am> instantiating exchange
am> instantiating environment
am> starting simulation
  
```

\* 1 iterations successfully completed

am> l en

#### ENTITY LIST

\*\*\*\*\*

entity user(\_U,\_S,\_D) { (3 parameters)

#### DEFINITION

number\_to\_dial[\_S,1] = \_D, tone[\_S,1] = |tone[\_S]|, ringing[\_S,1] = |ringing[\_S]|, onhook[\_S,1] = |onhook[\_S]|,  
 dialled\_number[\_S,1] = |dialled\_number[\_S]|, level[\_U,1] = 0, slow[\_U,1] = 0, only\_one[\_U,1] = 1,  
 selected[\_U,1] = 0

#### ACTION

(even&&ringing[\_S,1]) print(\_S," is ringing") -> ,

(even&&(slow[\_U,1]>0)) ->

slow[\_U,1] = (|slow[\_U,1]|-1),

(((((even&&

onhook[\_S,1])&&(level[\_U,1]==0))&&(only\_one[\_U,1]<=2))&&(selected[\_U,1]==0))&&(slow[\_U,1]==0)) -

```

>
    selected[_U,1] = 1 ;
    level[_U,1] = 1,
    (((even&&(selected[_U,1]==1))&&(level[_U,1]==1))&&(rand(3)==1)) print("replacing the receiver of ",_S) ->
    onhook[_S] = TRUE ;
    level[_U,1] = 2,
    ((even&&(selected[_U,1]==1))&&(level[_U,1]==2)) ->
    dialled_number[_S] = 0 ;
    level[_U,1] = 0 ;
    selected[_U,1] = 0,

    ((((((even&&!
    onhook[_S,1])&&(tone[_S,1]==1))&&(level[_U,1]==0))&&(only_one[_U,1]>=3))&&(only_one[_U,1]<=17))
    &&(selected[_U,1]==0))&&(slow[_U,1]==0)) ->
    selected[_U,1] = 2 ;
    level[_U,1] = 1,
    (((even&&(selected[_U,1]==2))&&(level[_U,1]==1))&&(rand(6)==1)) print("dialling the number ",_D," on
    telephone ",_S) ->
    dialled_number[_S] = number_to_dial[_S] ;
    level[_U,1] = 0 ;
    slow[_U,1] = 10 ;
    selected[_U,1] = 0,

    ((((((even&&!
    onhook[_S,1])&&(tone[_S,1]==4))&&(level[_U,1]==0))&&(only_one[_U,1]>=18))&&(only_one[_U,1]<=27)
    )&&(selected[_U,1]==0))&&(slow[_U,1]==0)) ->
    selected[_U,1] = 3 ;
    level[_U,1] = 1,
    (((even&&(selected[_U,1]==3))&&(level[_U,1]==1))&&(rand(6)==1)) print("speaking into ",_S) ->
    level[_U,1] = 0 ;
    selected[_U,1] = 0,

    ((((((even&&onhook[_S,1])&&!
    ringing[_S,1])&&(level[_U,1]==0))&&(only_one[_U,1]>=28))&&(only_one[_U,1]<=29))&&(selected[_U,1]=
    =0))&&(slow[_U,1]==0)) ->
    selected[_U,1] = 4 ;
    level[_U,1] = 1,
    ((even&&(selected[_U,1]==4))&&(level[_U,1]==1))&&(rand(3)==1)
    print("lifting up receiver of ",_S," to make call") ->
    onhook[_S] = FALSE ;
    level[_U,1] = 2,
    (((even&&(selected[_U,1]==4))&&(level[_U,1]==2))&&(rand(1)==1)) ->
    dial(_S,_D) ;
    level[_U,1] = 0 ;
    selected[_U,1] = 0,

    ((((((even&&onhook[_S,1])&&ringing[_S,1])&&(level[_U,1]==0))&&(only_one[_U,1]>=30))&&(only_one[
    _U,1]<=39))&&(selected[_U,1]==0))&&(slow[_U,1]==0)) ->
    selected[_U,1] = 5 ;
    level[_U,1] = 1,
    ((even&&(selected[_U,1]==5))&&(level[_U,1]==1))&&(rand(3)==1)
    print("lifting up receiver of ",_S," to answer") ->
    onhook[_S] = FALSE ;
    level[_U,1] = 2,
    (((even&&(selected[_U,1]==5))&&(level[_U,1]==2))&&(rand(6)==1)
    print("hello, ",_S,"") ->
    level[_U,1] = 0 ;
    selected[_U,1] = 0,
    !even -> tone[_S,1] = |tone[_S]|,
    !even -> ringing[_S,1] = |ringing[_S]|,
    !even -> onhook[_S,1] = |onhook[_S]|,
    !even -> dialled_number[_S,1] = |dialled_number[_S]|,
    (!even&&(selected[_U,1]==0)) ->
    only_one[_U,1] = |rand(50)|
}
2 instances

```

entity telephone(\_S,\_D) { (2 parameters)

DEFINITION

onhook[\_S] = TRUE, dialled\_number[\_S] = 0, connected[\_S,\_D,2] = |connected[\_S,\_D]|, connecting[\_S,\_D] = FALSE, engaged[\_S,\_D,2] = |engaged[\_S,\_D]|, dialling[\_S] = FALSE, tone[\_S] = if dialling[\_S] then 1 else if (connecting[\_S,\_D]&&engaged[\_S,\_D]) then 2 else if (connected[\_S,\_D]&&onhook[\_D]) then 3 else if ((connected[\_S,\_D]|connected[\_D,\_S])&&!onhook[\_D]) then 4 else 5

ACTION

!even -> connected[\_S,\_D,2] = |connected[\_S,\_D]|,  
!even -> engaged[\_S,\_D,2] = |engaged[\_S,\_D]|  
}

2 instances

entity dial(\_S,\_D) { (2 parameters)

DEFINITION

dialled\_number[\_S,3] = |dialled\_number[\_S]|, Tdial[\_S,3] = |Tdial|, time[\_S,3] = |time|, onhook[\_S,3] = |onhook[\_S]|, connecting[\_S,\_D,3] = |connecting[\_S,\_D]|, tstart[\_S] = |time|, valid[\_S] = (dialled\_number[\_S] != 0), dialling[\_S,3] = |(!onhook[\_S,3]&&((time[\_S,3]-tstart[\_S])<Tdial[\_S,3]))|, init\_flag[\_S,3] = TRUE, level[\_S,3] = 0, slow[\_S,3] = 0

ACTION

init\_flag[\_S,3] ->  
    dialling[\_S] = (!onhook[\_S]&&((time-tstart[\_S])<Tdial));  
    tstart[\_S] = |time|;  
    init\_flag[\_S,3] = FALSE,  
(even&&(slow[\_S,3]>0)) ->  
    slow[\_S,3] = (|slow[\_S,3]|-1),  
(((even&&(slow[\_S,3]==0))&&dialling[\_S,3])&&valid[\_S])&&(level[\_S,3]==0)) ->  
    level[\_S,3] = 1,  
(even&&(level[\_S,3]==1)) print("starting connection from ",\_S," to ",\_D) ->  
    connect(\_S,\_D);  
    level[\_S,3] = 0;  
    slow[\_S,3] = 20,  
(!even&&!init\_flag[\_S,3])&&!dialling[\_S,3]&&!connecting[\_S,\_D,3]) print("terminating dialler from ",\_S," to ",\_D) ->  
    dialling[\_S] = FALSE;  
    DELETE dial(\_S,\_D),  
(((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&(rand(2)==1))&&!init\_flag[\_S,3]) ->  
    dialled\_number[\_S,3] =  
|dialled\_number[\_S]|,  
(((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&(rand(10)==1))&&!init\_flag[\_S,3]) -> Tdial[\_S,3] =  
|Tdial|,  
(((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&!init\_flag[\_S,3]) -> time[\_S,3] = |time|,  
(((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&!init\_flag[\_S,3]) -> onhook[\_S,3] = |onhook[\_S]|,  
(((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&!init\_flag[\_S,3]) -> connecting[\_S,\_D,3] =  
|connecting[\_S,\_D]|, (((!even&&dialling[\_S,3])&&!connecting[\_S,\_D,3])&&!init\_flag[\_S,3]) ->  
    dialling[\_S,3] = |dialling[\_S]|  
}

0 instances

entity connect(\_S,\_D) { (2 parameters)

DEFINITION

onhook[\_S,\_S,4] = |onhook[\_S]|, onhook[\_S,\_D,4] = |onhook[\_D]|, ringing[\_D,4] = |ringing[\_D]|, Tcall[\_S,4] = |Tcall|, time[\_S,4] = |time|, tcall[\_S] = |time|, connected[\_S,\_D,4] = FALSE, answered[\_S,\_D,4] = FALSE, connecting[\_S,\_D,4] = !connected[\_S,\_D,4], engaged[\_S,\_D,4] = |engaged[\_S,\_D]|, init\_flag[\_S,4] = TRUE, level[\_S,4] = 0, only\_one[\_S,\_D,4] = 3, selected[\_S,\_D,4] = 0

ACTION

init\_flag[\_S,4] ->  
    answered[\_D] = FALSE;  
    tcall[\_S] = |time|;

```

connected[_S,_D] = FALSE ;
connecting[_S,_D] = !connected[_S,_D] ;
init_flag[_S,4] = FALSE,
((((even&&!engaged[_S,_D,4])&&!
connected[_S,_D,4])&&(only_one[_S,_D,4]==1))&&(selected[_S,_D,4]==0))&&(rand(3)==1))&&(level[_S,4]
==0)) ->
    level[_S,4] = 1 ;
    selected[_S,_D,4] = 1,
((even&&(level[_S,4]==1))&&(selected[_S,_D,4]==1)) print("connection made from ",_S," to ",_D) ->
    connected[_S,_D] = TRUE ;
    level[_S,4] = 0 ;
    selected[_S,_D,4] = 0,
((((even&&!onhook[_S,_D,4])&&!
answered[_S,_D,4])&&connected[_S,_D,4])&&(only_one[_S,_D,4]==2))&&(selected[_S,_D,4]==0))&&(level
[_S,4]==0)) ->
    level[_S,4] = 1 ;
    selected[_S,_D,4] = 2,
((even&&(level[_S,4]==1))&&(selected[_S,_D,4]==2)) print("registered picking up of receiver at ",_D) ->
    answered[_D] = TRUE ;
    level[_S,4] = 0 ;
    selected[_S,_D,4] = 0,
((((even&&engaged[_S,_D,4])&&(level[_S,4]==0))&&(only_one[_S,_D,4]==3))&&(selected[_S,_D,4]==0))
print("no connection possible from ",_S," to ",_D," because line is engaged") ->
    level[_S,4] = 1 ;
    selected[_S,_D,4] = 3,
((even&&(level[_S,4]==1))&&(selected[_S,_D,4]==3)) ->
    connecting[_S,_D] = FALSE ;
    connected[_S,_D] = FALSE ;
    DELETE connect(_S,_D),
(!even&&!init_flag[_S,4])&&!(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) print("connection from ",_S," to ",_D,"
broken") ->
    connected[_S,_D] = FALSE ;
    connecting[_S,_D] = FALSE ;
    DELETE connect(_S,_D),
(!even&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||ringing[_D])|
(connected[_S,_D]&&!answered[_D]||!onhook[_D])))&&(rand(2)==1))&&!init_flag[_S,4]) ->
    onhook[_S,_S,4] = |onhook[_S]|,

(!even&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||ringing[_D])|
(connected[_S,_D]&&!answered[_D]||!onhook[_D])))&&(rand(2)==1))&&!init_flag[_S,4]) ->
    onhook[_S,_D,4] = |onhook[_D]|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) ->    ringing[_D,4] = |ringing[_D]|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D])))&&(rand(10)==1)) ->
    Tcall[_S,4] = |Tcall|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) ->    time[_S,4] = |time|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D])))&&(rand(2)==1)) ->
    engaged[_S,_D,4] = |engaged[_S,_D]|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) ->    connected[_S,_D,4] = |
connected[_S,_D]|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) ->
    answered[_S,_D,4] = |answered[_D]|,
(!even&&!init_flag[_S,4])&&(!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))) ->
    connecting[_S,_D,4]=|connecting[_S,_D]|,
onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||ringing[_D])|(connected[_S,_D]&&!

```

```

answered[_D]||!onhook[_D]))))&&!engaged[_S,_D,4]&&!connected[_S,_D,4] ->
  only_one[_S,_D,4] = 1,
((((!even&&(selected[_S,_D,4]==0))&&!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))))&&!onhook[_S,_D,4]&&!
answered[_S,_D,4]&&connected[_S,_D,4]) ->  only_one[_S,_D,4] = 2,
(((!even&&(selected[_S,_D,4]==0))&&!onhook[_S]&&(((connecting[_S,_D]&&((time-tcall[_S])<Tcall))||
ringing[_D])|(connected[_S,_D]&&!answered[_D]||!onhook[_D]))))&&engaged[_S,_D,4]) ->
  only_one[_S,_D,4] = 3
}
0 instances
entity exchange() {      (0 parameters)
DEFINITION
onhook[6489,5] = |onhook[6489]|, onhook[7124,5] = |onhook[7124]|, connected[6489,7124] = FALSE,
connected[7124,6489] = FALSE, connecting[6489,7124,5]=|connecting[6489,7124]|,connecting[7124,6489,5]=|
connecting[7124,6489]|,answered[6489]=FALSE,answered[7124]=FALSE,ringing[7124]=((connected[6489,71
24]&&onhook[7124])&&!answered[7124]),ringing[6489]=((connected[7124,6489]&&onhook[6489])&&!
answered[6489]),engaged[6489,7124]=(connecting[6489,7124,5]&&(ringing[7124])|!
onhook[7124]),engaged[7124,6489]=(connecting[7124,6489,5]&&(ringing[6489])|!onhook[6489]),Tdial
=330,Tcall=340
ACTION
(!even&&(rand(2)==1)) ->
  onhook[6489,5] = |onhook[6489]|,
(!even&&(rand(2)==1)) ->
  onhook[7124,5] = |onhook[7124]|,
!even ->
  connecting[6489,7124,5]=|connecting[6489,7124]|,
!even ->
  connecting[7124,6489,5]=|connecting[7124,6489]|
}
1 instances
entity environment() {      (0 parameters)
DEFINITION
time = 0,level[6] = 0,even=0
ACTION
(even&&(level[6]==0)) ->
  level[6] = 1,
(even&&(level[6]==1)) ->
  time = (time+1) ;
  level[6] = 0,
TRUE ->
ev*****
*****
*****
*****
*****24] = 6489
Variable# 11: tone[7124,1] = 5
Variable# 12: ringing[7124,1] = FALSE
Variable# 13: onhook[7124,1] = TRUE
Variable# 14: dialled_number[7124,1] = 0
Variable# 15: level[11,1] = 0
Variable# 16: slow[11,1] = 0
Variable# 17: only_one[11,1] = 4
Variable# 18: selected[11,1] = 0
Variable# 19: onhook[6489] = TRUE
Variable# 20: dialled_number[6489] = 0
Variable# 21: connected[6489,7124,2] = FALSE
Variable# 22: connecting[6489,7124] = FALSE
Variable# 23: engaged[6489,7124,2] = FALSE
Variable# 24: dialling[6489] = FALSE
Variable# 25: tone[6489] = if dialling[6489] then 1 else if (connecting[6489,7124]&&engaged[6489,7124])
then 2 else if (connected[6489,7124]&&onhook[7124]) then 3 else if ((connected[6489,7124])

```



```

connected[7124,6489])&&!onhook[7124]) then 4 else 5
Variable# 26: onhook[7124] = TRUE
Variable# 27: dialled_number[7124] = 0
Variable# 28: connected[7124,6489,2] = FALSE
Variable# 29: connecting[7124,6489] = FALSE
Variable# 30: engaged[7124,6489,2] = FALSE
Variable# 31: dialling[7124] = FALSE
Variable# 32: tone[7124] = if dialling[7124] then 1 else if (connecting[7124,6489]&&engaged[7124,6489])
then 2 else if (connected[7124,6489]&&onhook[6489]) then 3 else if ((connected[7124,6489]||
connected[6489,7124])&&!onhook[6489]) then 4 else 5
Variable# 33: onhook[6489,5] = |onhook[6489]|
Variable# 34: onhook[7124,5] = TRUE
Variable# 35: connected[6489,7124] = FALSE
Variable# 36: connected[7124,6489] = FALSE
Variable# 37: connecting[6489,7124,5] = FALSE
Variable# 38: connecting[7124,6489,5] = FALSE
Variable# 39: answered[6489] = FALSE
Variable# 40: answered[7124] = FALSE
Variable#41:ringing[7124]=((connected[6489,7124]&&onhook[7124])&&!answered[7124])
Variable#42:ringing[6489]=((connected[7124,6489]&&onhook[6489])&&!answered[6489])
Variable#43:engaged[6489,7124]=((connecting[6489,7124,5]&&(ringing[7124])!|onhook[7124]))
Variable#44:engaged[7124,6489]=((connecting[7124,6489,5]&&(ringing[6489])!|onhook[6489]))
Variable# 45: Tdial = 330
Variable# 46: Tcall = 340
Variable# 47: time = 0
Variable# 48: level[6] = 0
Variable# 49: even = !0
      END OF DEFINITION STORE
      *****

```

am> 1 as

```

ACTION STORE
*****

```

```

Action# 1:(even&&ringing[6489,1]) print(6489," is ringing") ->
Action# 2:(even&&(slow[10,1]>0)) -> slow[10,1] = (|slow[10,1]|-1)
Action#3:((((even&&!
onhook[6489,1])&&(level[10,1]==0))&&(only_one[10,1]<=2))&&(selected[10,1]==0))&&(slow[10,1]==0)) -
>
      selected[10,1] = 1 ;
      level[10,1] = 1
Action#4:(((even&&(selected[10,1]==1))&&(level[10,1]==1))&&(rand(3)==1)) print("replacing the receiver
of ",6489) ->
      onhook[6489] = TRUE ;
      level[10,1] = 2

Action# 5:((even&&(selected[10,1]==1))&&(level[10,1]==2)) ->
      dialled_number[6489] = 0 ;
      level[10,1] = 0 ;
      selected[10,1] = 0
Action#6:((((((even&&!
onhook[6489,1])&&(tone[6489,1]==1))&&(level[10,1]==0))&&(only_one[10,1]>=3))&&(only_one[10,1]<=1
7))&&(selected[10,1]==0))&&(slow[10,1]==0)) ->
      selected[10,1] = 2 ;
      level[10,1] = 1
Action#7:(((even&&(selected[10,1]==2))&&(level[10,1]==1))&&(rand(6)==1)) print("dialling the number
",7124," on telephone ",6489) ->
      dialled_number[6489]=number_to_dial[6489] ;
      level[10,1] = 0 ;
      slow[10,1] = 10 ;
      selected[10,1] = 0
Action#8:((((((even&&!
onhook[6489,1])&&(tone[6489,1]==4))&&(level[10,1]==0))&&(only_one[10,1]>=18))&&(only_one[10,1]<=

```

```

27))&&(selected[10,1]==0)&&(slow[10,1]==0)) ->
    selected[10,1] = 3 ;
    level[10,1] = 1
Action#9:(((even&&(selected[10,1]==3))&&(level[10,1]==1))&&(rand(6)==1)) print("speaking into ",6489) -
>
    level[10,1] = 0 ;
    selected[10,1] = 0
Action#10:((((even&&onhook[6489,1])&&!
ringing[6489,1])&&(level[10,1]==0)&&(only_one[10,1]>=28)&&(only_one[10,1]<=29)&&(selected[10,1]=
=0)&&(slow[10,1]==0)) ->
    selected[10,1] = 4 ;
    level[10,1] = 1
Action#11:(((even&&(selected[10,1]==4))&&(level[10,1]==1))&&(rand(3)==1)) print("lifting up receiver of
",6489," to make call") ->
    onhook[6489] = FALSE ;
    level[10,1] = 2
Action#12:(((even&&(selected[10,1]==4))&&(level[10,1]==2))&&(rand(1)==1)) ->
    dial(6489,7124) ;
    level[10,1] = 0 ;
    selected[10,1] = 0
Action#13:
((((((even&&onhook[6489,1])&&ringing[6489,1])&&(level[10,1]==0)&&(only_one[10,1]>=30)&&(only_o
ne[10,1]<=39))&&(selected[10,1]==0)&&(slow[10,1]==0)) ->
    selected[10,1] = 5 ;
    level[10,1] = 1
Action#14:(((even&&(selected[10,1]==5))&&(level[10,1]==1))&&(rand(3)==1)) print("lifting up receiver of
",6489," to answer") ->
    onhook[6489] = FALSE ;
    level[10,1] = 2
Action#15:(((even&&(selected[10,1]==5))&&(level[10,1]==2))&&(rand(6)==1)) print("hello, ",6489,"") ->
    level[10,1] = 0 ;
    selected[10,1] = 0
Action# 16:!even ->
    tone[6489,1] = |tone[6489]
Action# 17:!even ->
    ringing[6489,1] = |ringing[6489]
Action# 18:!even ->
    onhook[6489,1] = |onhook[6489]
Action# 19:!even ->
    dialled_number[6489,1]=|dialled_number[6489]
Action# 20:(!even&&(selected[10,1]==0)) ->
    only_one[10,1] = |rand(50)
Action# 21:(even&&ringing[7124,1]) print(7124," is ringing") ->
Action# 22:(even&&(slow[11,1]>0)) ->
    slow[11,1] = (|slow[11,1]-1)
Action#23:((((even&&!
onhook[7124,1])&&(level[11,1]==0)&&(only_one[11,1]<=2))&&(selected[11,1]==0)&&(slow[11,1]==0)) ->
    selected[11,1] = 1 ;
    level[11,1] = 1
Action#24:(((even&&(selected[11,1]==1))&&(level[11,1]==1))&&(rand(3)==1)) print("replacing the receiver
of ",7124) ->
    onhook[7124] = TRUE ;
    level[11,1] = 2
Action# 25:(even&&(selected[11,1]==1))&&(level[11,1]==2)) ->
    dialled_number[7124] = 0 ;
    level[11,1] = 0 ;
    selected[11,1] = 0
Action#26:((((((even&&!
onhook[7124,1])&&(tone[7124,1]==1))&&(level[11,1]==0)&&(only_one[11,1]>=3))&&(only_one[11,1]<=17
))&&(selected[11,1]==0)&&(slow[11,1]==0)) ->
    selected[11,1] = 2 ;
    level[11,1] = 1
Action#27:(((even&&(selected[11,1]==2))&&(level[11,1]==1))&&(rand(6)==1)) print("dialling the number

```

```

",6489," on telephone ",7124) ->
    dialled_number[7124]=number_to_dial[7124] ;
    [11,1] = 0 ;
    slow[11,1] = 10 ;
    selected[11,1] = 0
Action#28:((((even&&!
onhook[7124,1])&&(tone[7124,1]==4))&&(level[11,1]==0))&&(only_one[11,1]>=18))&&(only_one[11,1]<=2
7))&&(selected[11,1]==0))&&(slow[11,1]==0)) ->
    selected[11,1] = 3 ;
    level[11,1] = 1
Action#29:(((even&&(selected[11,1]==3))&&(level[11,1]==1))&&(rand(6)==1)) print("speaking into ",7124) -
>
    level[11,1] = 0 ;
    selected[11,1] = 0
Action#30:((((even&&onhook[7124,1])&&!
ringing[7124,1])&&(level[11,1]==0))&&(only_one[11,1]>=28))&&(only_one[11,1]<=29))&&(selected[11,1]=
=0))&&(slow[11,1]==0)) ->
    selected[11,1] = 4 ;
    level[11,1] = 1
Action#31:(((even&&(selected[11,1]==4))&&(level[11,1]==1))&&(rand(3)==1)) print("lifting up receiver of
",7124," to make call") ->
    onhook[7124] = FALSE ;
    *****
    *****
    *****
    *****
    *****
    *****= 1
Action#34:(((even&&(selected[11,1]==5))&&(level[11,1]==1))&&(rand(3)==1)) print("lifting up receiver of
",7124," to answer") ->
    onhook[7124] = FALSE ;
    level[11,1] = 2

Action#35:(((even&&(selected[11,1]==5))&&(level[11,1]==2))&&(rand(6)==1)) print("hello, ",7124,"") ->
    level[11,1] = 0 ;
    selected[11,1] = 0
Action# 36:!even ->
    tone[7124,1] = |tone[7124]|
Action# 37:!even ->
    ringing[7124,1] = |ringing[7124]|
Action# 38:!even ->
    onhook[7124,1] = |onhook[7124]|
Action# 39:!even ->
    dialled_number[7124,1]=|dialled_number[7124]|
Action# 40:(!even&&(selected[11,1]==0)) ->
    only_one[11,1] = |rand(50)|
Action# 41:!even ->
    connected[6489,7124,2]=|connected[6489,7124]|
Action# 42:!even ->
    engaged[6489,7124,2]=|engaged[6489,7124]|
Action# 43:!even ->
    connected[7124,6489,2]=|connected[7124,6489]|
Action# 44:!even ->
    engaged[7124,6489,2]=|engaged[7124,6489]|
Action# 45:(!even&&(rand(2)==1)) ->
    onhook[6489,5] = |onhook[6489]|
Action# 46:(!even&&(rand(2)==1)) ->
    onhook[7124,5] = |onhook[7124]|
Action# 47:!even ->
    connecting[6489,7124,5]=|connecting[6489,7124]|
Action# 48:!even ->
    connecting[7124,6489,5]=|connecting[7124,6489]|
Action# 49:(even&&(level[6]==0)) ->

```

```
level[6] = 1
Action# 50:(even&&(level[6]==1)) ->
time = (|time|+1) ;
level[6] = 0
Action# 51:TRUE ->
even = !|even|
END OF ACTION STORE
*****
```

```
am> l in          list the currently instantiated entities
INSTANCES
*****
```

```
user (10,6489,7124)
user (11,7124,6489)
telephone (6489,7124)
telephone (7124,6489)
exchange ()
environment ()
END OF INSTANCES
```

```
am> status
nflag = TRUE
aflag = TRUE
silent = FALSE
iterations = 1
am> set iterations = 100
```

```
am> cont
continuing simulation
```

```
(40) lifting up receiver of 7124 to make call
(64) dialling the number 6489 on telephone 7124
(68) starting connection from 7124 to 6489
(73) terminating dialler from 7124 to 6489
(76) connection made from 7124 to 6489
(78) 6489 is ringing
(80) 6489 is ringing
(82) 6489 is ringing
(84) 6489 is ringing
(86) 6489 is ringing
(86) lifting up receiver of 6489 to answer
(94) registered picking up of receiver at 6489
(100) 'hello, 6489'
```

```
* 100 iterations successfully completed
```

```
am> l in
INSTANCES
*****
user (10,6489,7124)
user (11,7124,6489)
telephone (6489,7124)
telephone (7124,6489)
connect (7124,6489)          7124 is currently connected to 6489
exchange ()
environment ()
END OF INSTANCES
```

```
am> load runset
loading run set
#
* loaded run set
am> l runset
```

```
RUN SET
*****
```

```
level[6] = 1 ;          no telephone will be put down...  
even = !TRUE
```

```
    END OF RUN SET  
*****
```

```
am>?(only_one[10,1])  
only_one[10,1] is defined as 16    because only_one[10,1] has the wrong value  
only_one[10,1] evaluates to 16  
am> define only_one[10,1] = 1 ;    so set it to a value...  
defining only_one[10,1]  
am> status  
nflag = TRUE  
aflag = TRUE  
silent = FALSE  
iterations = 100  
am> set iterations = 6  
am> cont  
continuing simulation
```

```
(106) replacing the receiver of 6489    ...that will cause the "replace receiver"  
      guard to be evaluated
```

```
(107) connection from 7124 to 6489 broken
```

```
* 6 iterations successfully completed
```

```
am>
```

```
script done on Mon Aug 28 18:52:08 1989
```