Experiments with a Room

You should first make a copy of the room.d script from /dcs/acad/wmb/public/demo/DONALD into your own file space. Run that with the command

```
cat room.d - | donald | eden -n
```

and check that you can then make redefinitions to the prompt D[/]> and that these seem to work properly.

Experiment 1:

Change the size of the room and the sizes of the furniture. Note some other dependent dimensions also change (e.g. the drawer dimensions depend on those of the desk). Move the furniture, note the need for knowledge of the 'preferred' reference corner of the table and desk. Try changing the reference point for the table to the centre of the table. Is this better? Note that, in this script, you can perform 'unintended' distortions of the furniture and room.

Experiment 2:

Add a new feature to your room, preferably one that has a simple 'behaviour' associated with it—perhaps a window, or a bookcase which is concealing a secret door etc. Or make existing features such as the door and desk drawer behave more realistically.

Experiment 3:

In planning a realistic amount of space for the desk it might be convenient to include (say) a chair and a waste bin which are to be always included near the desk. Put these within the openshape desk definition or, if you prefer, put the desk, chair and bin all within a new openshape 'workspace'. Try moving this combination as a whole. How could you rotate it to put it against an adjacent wall? Arrange for the bin to be at the 'right(left)-hand' end of the desk according to whether the user is right (left)-handed.

Experiment 4:

What counts as *appropriate* transformations of state (changes in the displayed room in this case), and the redefinitions which produce them, often depends on the intended user. Most users can open or close a door, but the architect may put the door in a different position, change width, make it slide instead of hinge etc. Write a suitable script for an 'architect's door' which allows such changes.

MSC/S4
SBR 14/12/92

Getting Started with DoNaLD

In order to use definitive scripts written in your own directory you need to have the system variable PUBLIC suitably defined in your .cshrc (or .profile or equivalent). A recipe for doing this which can be cut and pasted will be found in /dcs/acad/wmb/public/MSC.README. (That file is also the main reference for information about definitive notations for use on this module. Simon Yung will be maintaining this file and will be able to explain things in there further if they are not already clear. It will also be most useful to tell him of things which you find are needed but missing.)

Set-up for initial DoNaLD Demo

The initial DoNaLD Demo will be a passive demonstration of a particular script (room.d) in order to introduce the DoNaLD syntax. To follow this demo it will be useful to open three windows and type the following in them:

xterm1:

jove (or your favourite editor) /dcs/acad/wmb/public/demo/DONALD/room.d

This is the script that is being demonstrated and modified. Hard copy will also be available.

xterm2:

cd /dcs/acad/wmb/public/demo

demo.donald room

This will run the script room.d and display the current state of the room. Here also is the DoNaLD prompt to which you may type (or paste) redefinitions in order to change the state of the room.

xterm3:

cd /dcs/acad/wmb/public/demo

tail -f don.log

This should be displaying the redefinitions, or other commands being demonstrated. It should be possible to cut them out of here and paste into the DoNaLD prompt in your *xterm2*.

The first half page or so of the script room.d, the definitions of the walls of the room and of the door, illustrate most of the syntax you will need to begin with. More comprehensive details may be found in the file MSC.README.

Notice that the order of definitions (or redefinitions) does not matter. The current script (i.e. the 'latest' set of definitions for the variables) corresponds at any stage to the state of the room as displayed. It may sometimes be necessary to 'refresh' the screen in order to have that state properly displayed. The script represents a rather primitive sort of room in that the 'objects' only resemble objects in a small number of ways. Bizarre things can easily happen. The following experiments with this environment should help you feel more at home in the room and with the DoNaLD notation.