

---

## Using definitive notations to represent state

---

### Observation and action associated with a definitive script

When interpreting a definitive script, you should keep in mind the entire family of possible redefinitions and new definitions. This has been illustrated via the room model in Lab 1, where the range of possible definitions includes definitions that correspond (in some informal metaphorical way) to

- sensible ordinary actions that a user might take (opening and shutting a door, relocating a table, introducing a new piece of furniture etc),
- actions that a builder or carpenter might be able to take, such as making the door wider, extending the room
- extensions that a designer or architect might make in order to make the room model more useful, allowing items of furniture to be moved conveniently through graphical user interaction, monitoring whether the cable is long enough, or the table obstructs the door.
- actions that undermine the integrity of the room model, making what seem to be obvious or established interpretations problematic, as when turning the outline of the table into an incoherent bunch of lines, or potentially overturning the whole idea that the line drawing refers to a room at all.

Note how different possible redefinitions are associated with setting up different modes and contexts for observation, with actions on the part of different agents, and with actions related to different motivations. However hard you try to classify the possible modes of observation and kinds of agency that might be relevant to the room model, you will never succeed. The model may be quite crude and simple, but its possible meanings cannot be circumscribed, in much the same way that each of us individually makes something different of their experience even of what appears to be a common object of attention. What we can do is to build up a family of standard interactions that is associated with each particular situation and motivation. We regard this as *modelling* because it relies upon establishing a correlation between the experience offered by the computer and some external experience moment by moment ... and thus is (as if) carried out in a situation in which there is a referent. In fact, "models" and "referents" in a more conventional sense only emerge from patterns of interaction and interpretation that become stable.

A primary difference between Empirical Modelling and programming is that it puts the emphasis on setting up states for all kinds of potential ways of observing and acting, rather than focusing on specific types of agency, goals and actions. This is what we are describing as "modelling state-as-experienced". What does "modelling state-as-experienced" mean? The point has just been made that what is there to be experienced in any state is much more than can be comprehensively apprehended and set down. It depends on who you are and what sort of privileges to change state you have (a room user can't move a wall etc). It depends on your imagination and the purposes you have in view. It depends on your degree of familiarity with the situation and level of acquired understanding and skill. You could revisit a room in "the same state" at different stages (e.g. as you encounter a room first as a baby, and a little later in your life as a small child) and have developed quite a different perception. Two points are particularly significant here:

- **The script is not a static declarative object**

To appreciate the distinctive qualities of modelling with definitive scripts you need to consider not just the script as it is at this moment but the extensible evolving script and family of interactions and interpretations that develop from it over time. In this evolution, some features stabilise, and others dissolve. You can think of formal objects like programs as crystallizing as patterns out of these interactions.

- **It's not what the script represents, but how it relates to what it doesn't (yet) represent**

What makes the script particularly significant as a medium is the fact that reconfiguring dependency relationships is a fundamental element of sense-making. By modifying the definitions in a script, you can conveniently accommodate shifts in perception. As a simple illustration, we can introduce a definition to the room script when we first connect "opening the door" with "the light coming on".