**Actor:**

Someone or something that interacts with the system. Could be a person, could be computer, could even be fucking dog.

**Usage pattern:**

Seems to be some way of mapping actor-system interaction and all the resulting results that result from it.

**Procedure pattern**:

Describes a process wherein is sequentially transitioning from state to state. F.eks., a user can insert a card into an ATM, then enter their code, if it matches they are shown some values they can choose from, and then they can either try and withdraw it or cancel altogether.

**Material pattern:**

The material pattern exists in stark contrast to the Procedure pattern, as instead of sequentially transitioning through different states, it has a main/general state in which the system will exist on for most of the programs running time. From this general state, the user can perform actions which changes the system’s state. If the state the program changes to is not another general state, it will usually return back to the state before it.

An example of such a program is word! After a document has been opened, the user can position the text cursor somewhere, which changes the system to the general state of *Cursor placed*. From this state, the user can then either place the cursor somewhere else or enter text, both of which are actions where the system state does not change. Alternatively, the user can from her mark some text, changing the state to *Text Selected*. From here, the user can change text fonts without changing the state, or delete/overwrite the marked text, changing the system state back to the general state.

**Quiz corrections:**









