PAT 451/551 INTERACTIVE MEDIA DESIGN I

VERPLANK'S INTERACTION DESIGN FRAMEWORK

BILL VERPLANK

Pioneer of interaction design; mechanical engineer by training

Worked for Xerox, IDTwo, IDEO, and Interval Research—all places that were instrumental in developing design research and interaction design practice

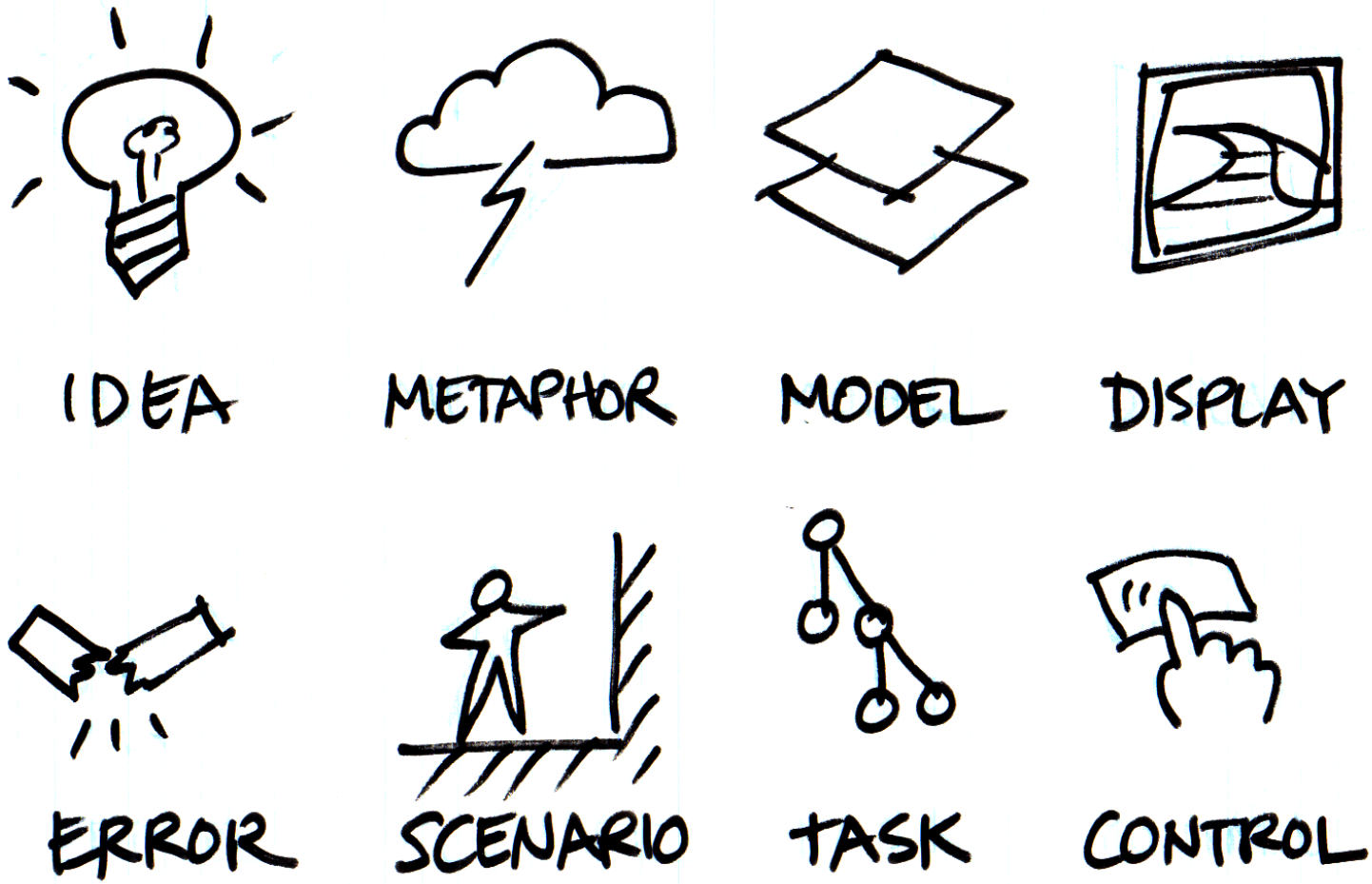
Through interactions with Max Mathews at Interval, became interested in interaction design as applied to music

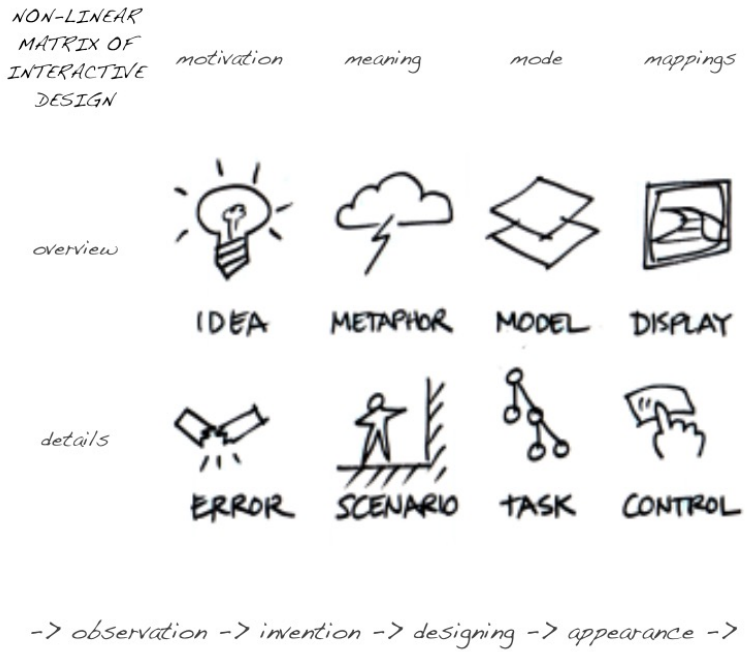
BILL VERPLANK

For more, see

- Canvas::Files>Articles>verplank_IDsketchbook.pdf:
  - https://umich.instructure.com/courses/387183/files/folder/Articles?preview=15703105
- http://www.designinginteractions.com/interviews/BillVerplank
- https://vimeo.com/20285615

Verplank's Interaction design framework



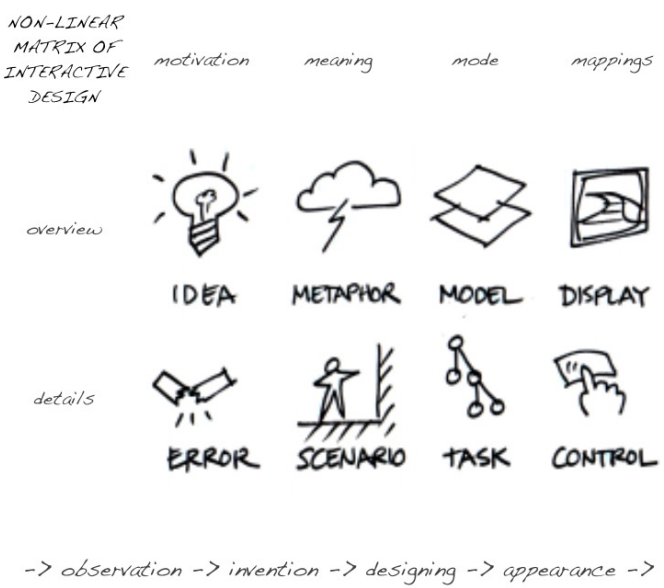


The framework is intended to be a “non-linear” guide to the considerations for producing a complete interaction design.

Non-linear: you can start anywhere.

Complete design: unless you’ve considered each of these aspects, something will be missing.

The innovation here is that, contrary to the way design tends to be discussed and taught, Verplank argues that there is no singular process or way of approaching design.



The top row represents “overviews” – big-picture considerations.

The bottom row represents “details”.

Vertical columns are conceptually paired: idea and error are counterparts.

Moving left to right, the columns represent:

OBSERVATION INVENTION DESIGN APPEARANCE

So although Verplank says you can START your design anywhere, the layout, organization, and relationships of the 8 squares are fixed. (ERROR would never be paired with SCENARIO, for example.)

Conventional wisdom about design says you should start with an ERROR.

ERROR is the problem your design is trying to solve. Verplank puts it under the category of “Observations” or ”Motivations”. This is the thing you’ve observed in the world that is broken or wrong, that needs to be improved or fixed; the motivation for making something new.

NOTE that this is the biggest mistake students tend to make. The ERROR is NOT the problem with your design. It is the problem in the world that your design tries to solve.

The IDEA is the big-picture description of the design. The “AHA!” Think of it as the description of the design in around 10 words

“*Bathroom faucets that you don’t have to touch!*”

“*Don’t pour your soup in the BLENDER; put your blender in the SOUP!*”

The SCENARIO is straightforward. It is an account of the situation in which the design will be used. Who is the user? Where are they? What else is going on around them that is relevant?

METAPHOR

The METAPHOR is sometimes tricky to grasp.

The METAPHOR is the answer to the question: what other, familiar, real-world activities and objects do we leverage to help the user understand the design and how it works.

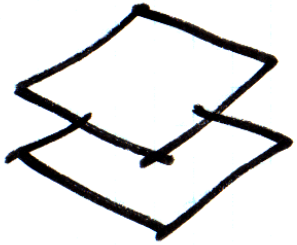
Think about the METAPHORS we use in our computer, for example. We talk about a DESKTOP, with FILES and FOLDERS. FILES are stored in FOLDERS. We put things in the TRASH. We can remove them from the TRASH until we EMPTY it, after which they are gone.

None of these describes or is required by the actual underlying design of the file system of your computer. We could just as easily call what we know as “files” “MARBLES” and call what we know as “folders” CUPS. We could talk about tossing a marble into a cup to store it.

METAPHOR

Notice how the set of metaphors in Desktop Computing gives USERS familiar concepts to understand how we can interact with the system, but it also allows DESIGNERS to make the interface consistent and cohesive.

Many design decisions will follow logically or thematically from the metaphor you choose.

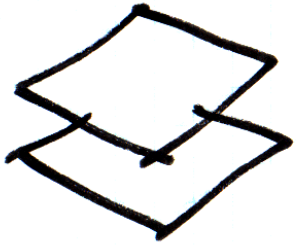


MODEL

The MODEL is “what the user needs to know or understand to successfully use the system.” It is sometimes useful to think about it in terms of “what would the user need to know to recover from a breakdown or failure of the system?”

Think about a mobile phone. What if you unlock your phone, dial a number, press “Call” and nothing happens? What do you need to know or understand to recover from error? Well, you need some idea that your phone sends and receives wireless signals to a device on a tower or a rooftop somewhere (“the network!”), and it needs to be close enough to one of those devices in order to make a call.

To use a mobile phone, you also need to understand that your everyone’s phone has a unique identifier that distinguishes it from all others and allows your call to only be received by that phone.

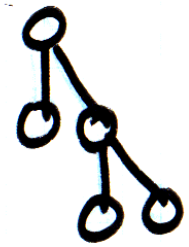


MODEL

The MODEL of a vending machine might be:

* Each item has an identifying code and a price
* You put in the appropriate amount of money for the item you want, and then enter the code for that item
* The machine will only dispense the item if you've put in at least as the correct amount of money for that item
* If you've put in more money than the cost of that item, the machine will give you change.

(It is your job to represent that iconically, in a sketch!)



TASK

The TASK is relatively straightforward. It is the set of steps that the user is expected to carry out. Think of it as the ‘implementation’ of the MODEL.

So, for the previous vending machine example, the TASK might be:

1. Locate the item you want.
2. Insert money into the slot
3. Enter the code for the item you want on the keypad.
4. Receive item at the bottom of the machine.
5. Take your change from the change tray.

DISPLAY

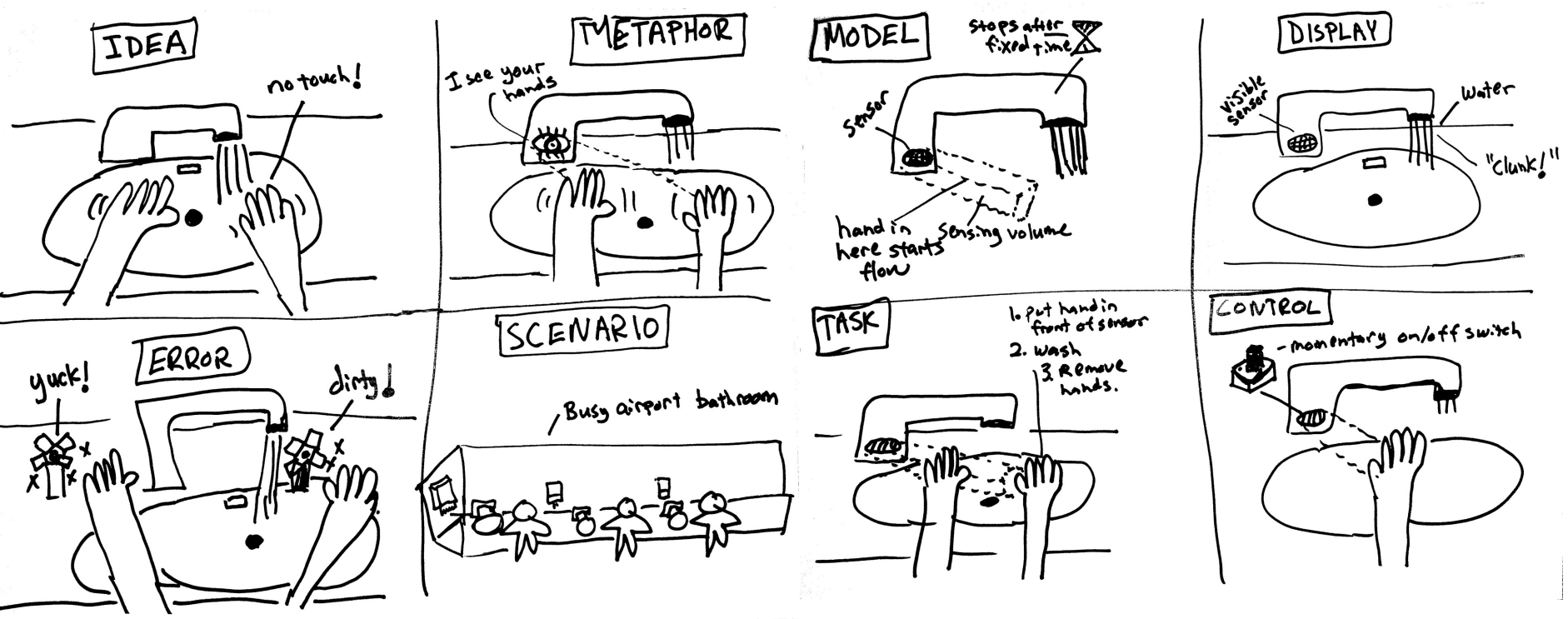
The DISPLAY is what the user sees/hears/feels (or even tastes/smells). It is what the system provides to us as feedback to help us understand how to interact with it.

Think about the card reader on the door to the Workshop. It has a red light that shows that the door is locked. When you present your card, the red light goes off and the green light goes on. There is an LCD display that tells you “Access Granted.” You also hear a “click” as the solenoid disengages the lock.

CONTROL

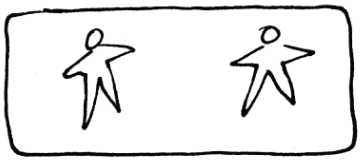
The CONTROL is what the user is provided to physically interact with. What do they DO? This is where the designer decides whether they should have discrete (buttons) or continuous (sensors) controls. How many? If the interface has continuous controls, what is the dynamic range or number of steps/values?

ANALYTICAL EXAMPLE



A GENERATIVE EXAMPLE FROM VERPLANK

IDEA



one-to-one silent and personal.

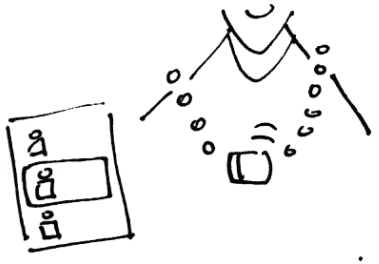
METAPHOR

Holding hands give a squeeze

MODEL

1. Single Channel
2. Packets

DISPLAY



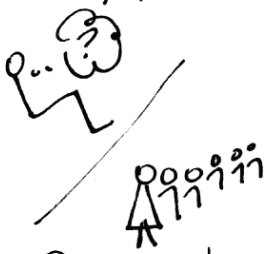
List on Pilot

Vibration on neclace

ERROR

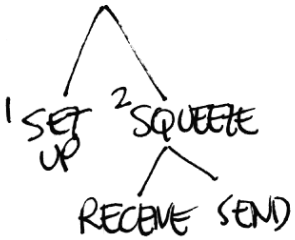
Cell phones ring in public.

SCENARIO



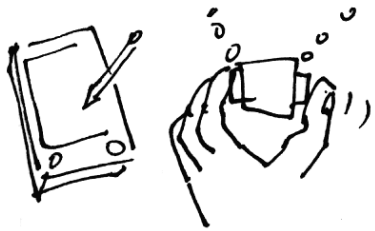
Sam at home Sally stuck at checkout

TASK



1. SET UP Sam/Sally
2. SQUEEZE

CONTROL



SELECT (tap)

SQUEEZE