

The nature of design

Prof. JIGISHA TRIVEDI
INFORMATION TECHNOLOGY
SPCE

The nature of design

- The entire purpose of creativity can be summarized as manipulations aimed at achieving asymmetric outcomes.
- For example, from combinatory processes we get emergent properties — the whole being greater than the sum of the parts — meaning that value has been created.
- (A specific example of combinatory creation can be found in the addition of the electric motor to a washbasin to create the washing machine.)
- However, there is a subtle but powerful aspect of creativity that often goes overlooked: let's call it “alteration.”

- By “alteration” I mean a reorganizing of elements, as opposed to the addition or substitution of elements.
- When it comes to software, tweaks in the user interface (UI) can often be viewed in this light, and I’d attribute the success of the iPhone, for example, to be in no small part due to its UI.
- The iPhone’s UI is so intuitive I’ve seen multiple infants use it with little or no prior exposure.
- These examples of “alteration” represent a low-cost way to create enormous added value. For every element and combination of elements, there is an ideal form it can assume — ideal in regards to some objective.



Transforming the World

- Every conceivable organization, invention, life-form, mode of behavior, written work, etc., could see a dramatic increase in value by moving closer to this ideal form in respect to some objective.
- And, lest we consider hands and plates to be just another type of foot or bowl, then something new has been created through this process.
- This creativity through transformation, or alteration for optimization, underlies the purpose of design. **Design can make a thing more beautiful, usable, or powerful without additives.**

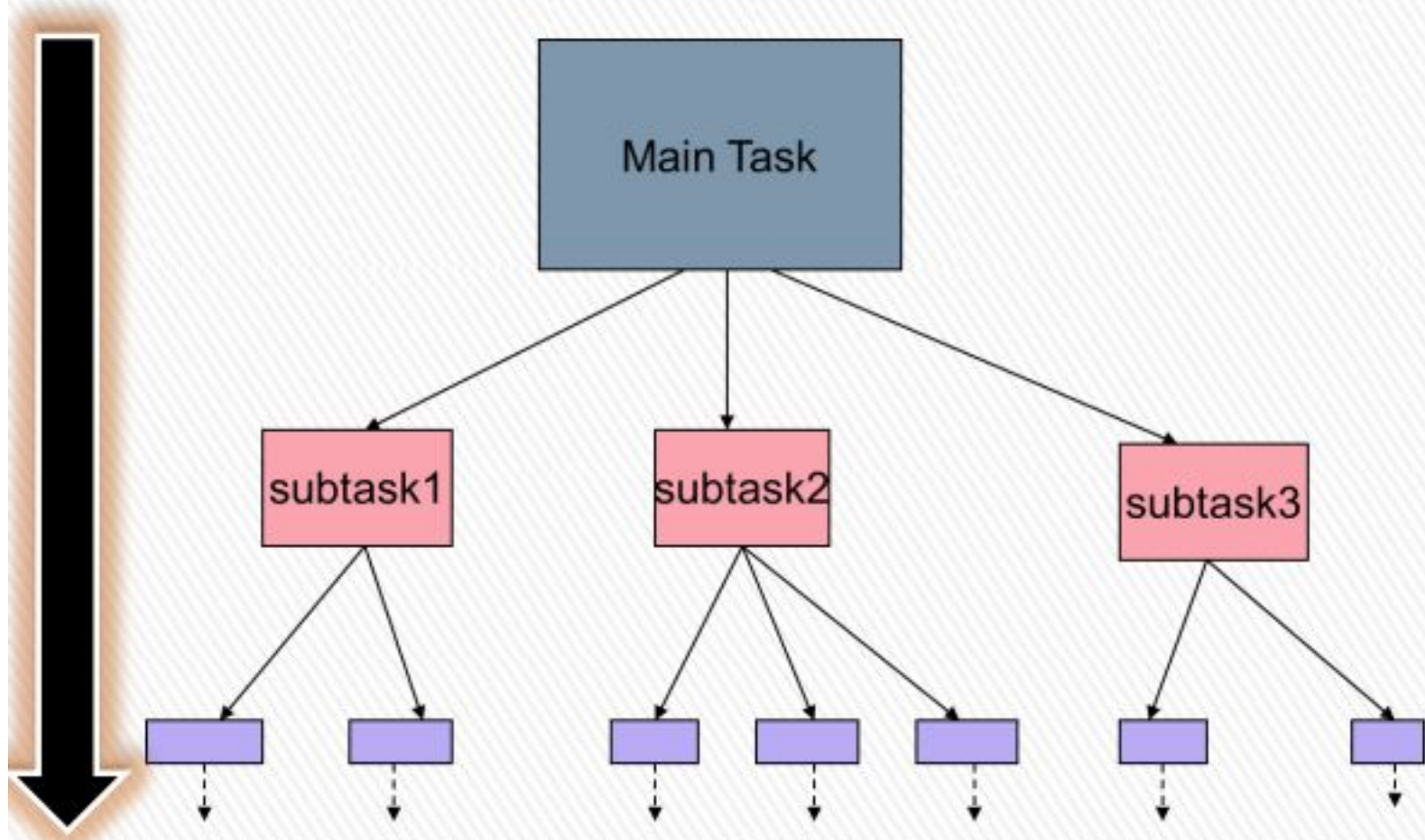
Top Down Design Model

- In top-down model, an overview of the system is formulated, without going into detail for any part of it.
- Each part of the system is then refined in more details.
- Each new part may then be refined again, defining it in yet more details until the entire specification is detailed enough to validate the model.

Top Down Concept in Problem Solving

- » If we look at a problem as a whole, it may seem impossible to solve because it is so complex.
- Examples:
 - writing a University System program
 - writing a word processor
- Complex problems can be solved using topdown design, also known as stepwise refinement, where
 - > We break the problem into parts
 - > Then break the parts into parts
 - > Soon, each of the parts will be easy to do

Top Down Design

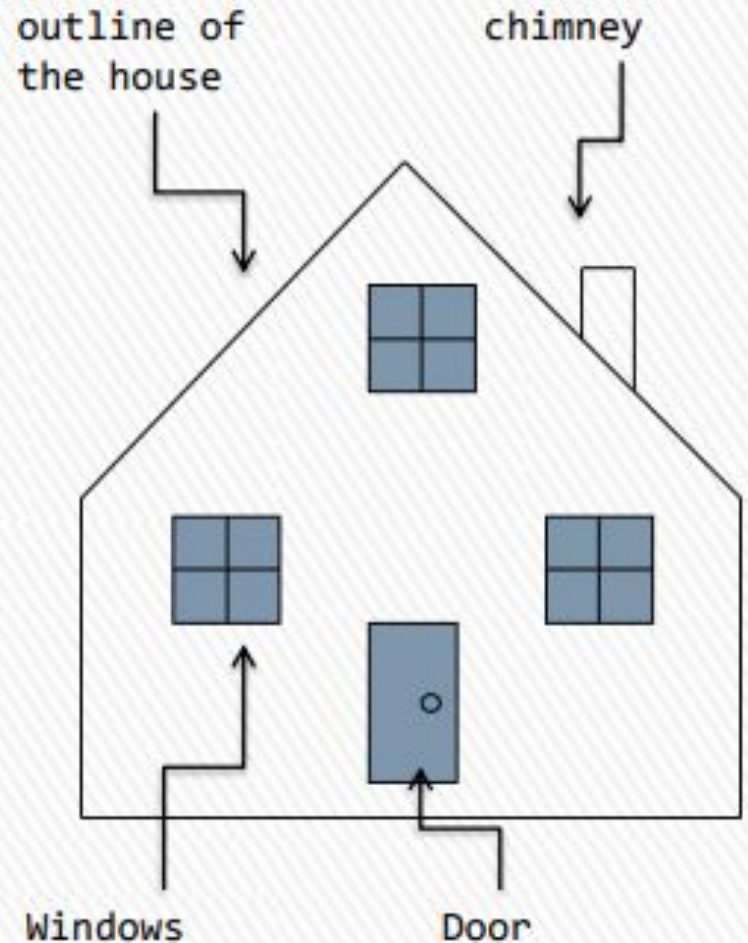


Advantages of Top-Down Design

- Breaking the problem into parts helps us to clarify what needs to be done.
- At each step of refinement, the new parts become less complicated and, therefore, easier to figure out.
- Parts of the solution may turn out to be reusable.
- Breaking the problem into parts allows more than one person to work on the solution

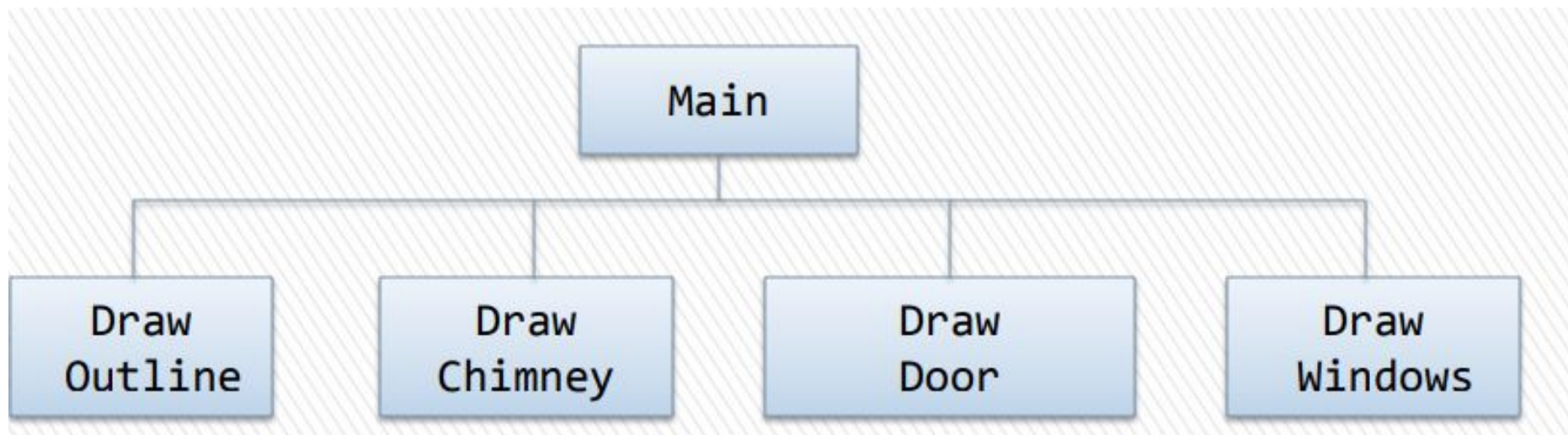
Example

- » Problem:
- » Write a program that draws this picture of a house.



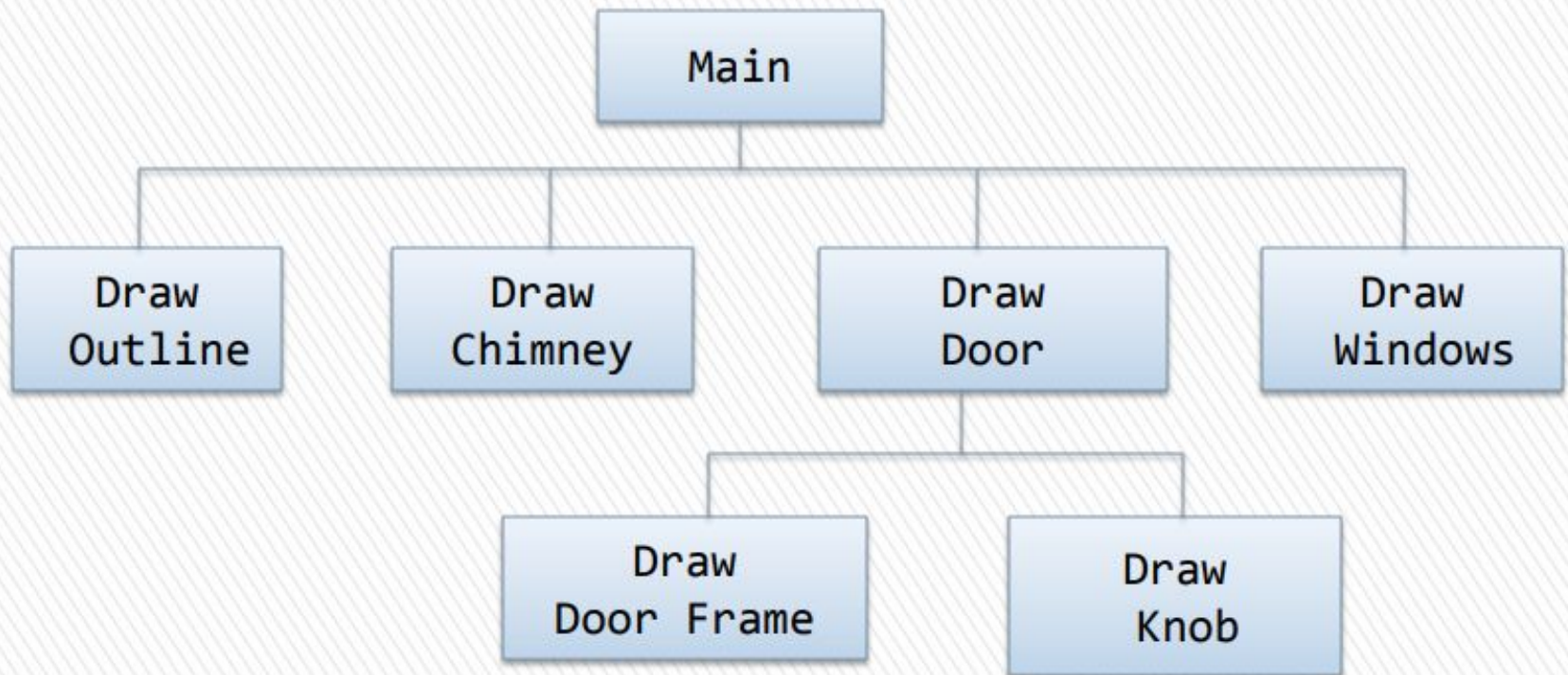
The Top Level

- Draw the outline of the house
- Draw the chimney
- Draw the door
- Draw the windows



Observation

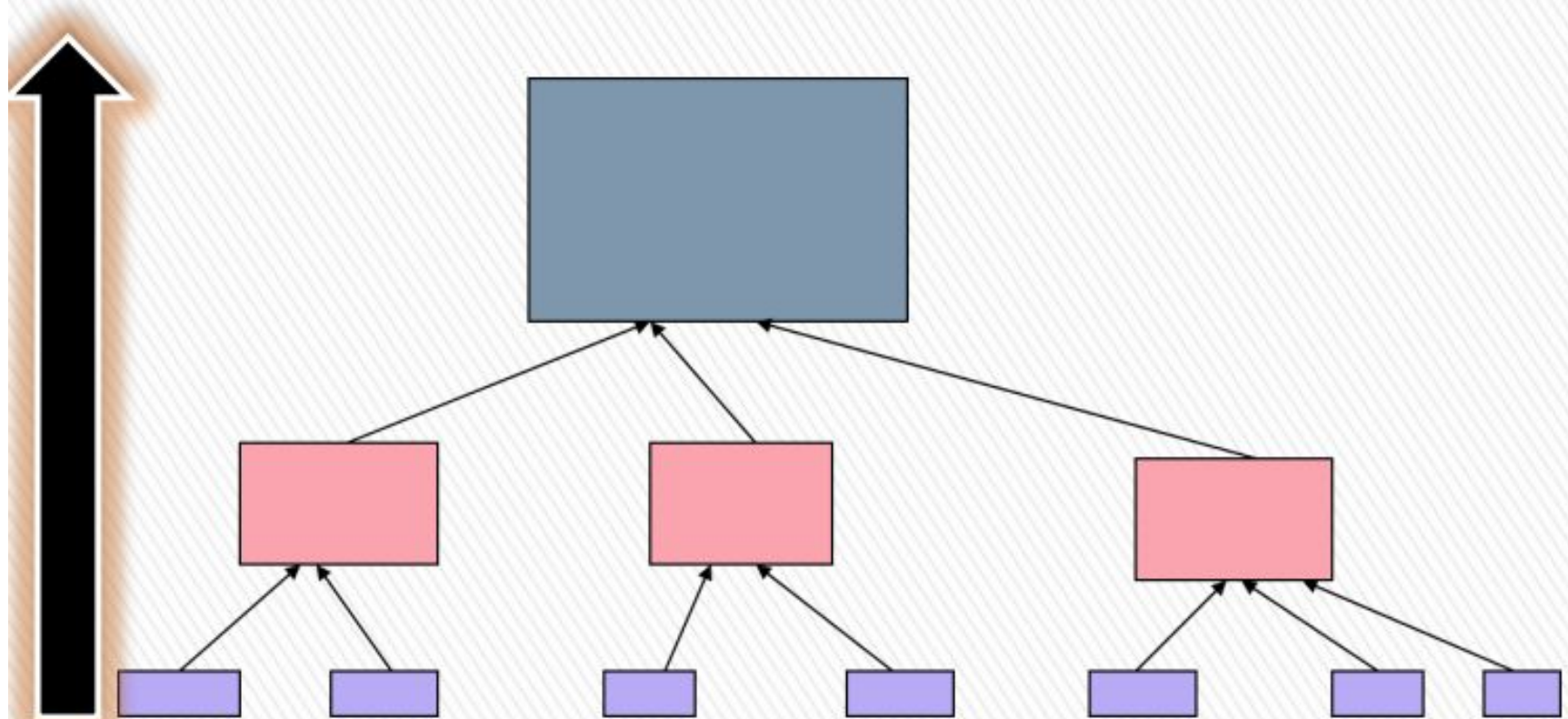
- » The door has both a frame and knob. We could break this into two steps.

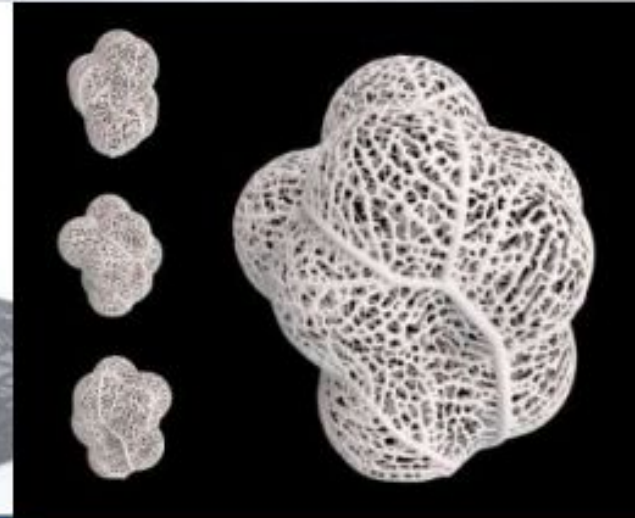
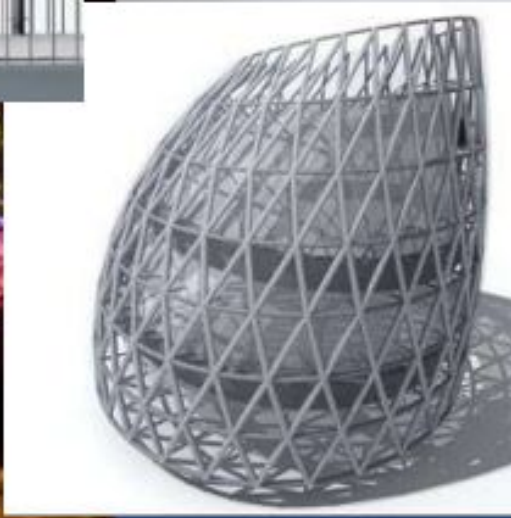
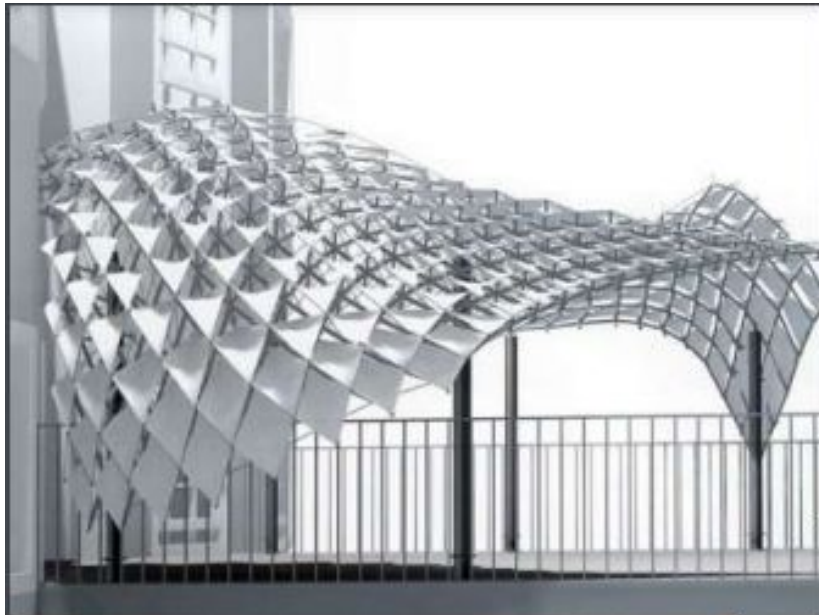


Bottom-up Design

- In bottom-up design individual parts of the system are specified in details.
- The parts are then linked together to form larger components, which are in turn linked until a complete system is formed.
- Object-oriented languages such as C++ or JAVA use bottom-up approach where each object is identified first.

Bottom up Design





DESIGN

Generative Design

- Generative Design, in particular, an algorithmic approach, is an efficient, rigorous, controllable, and flexible tool that allows, with reduced time and effort, the production of several different models to explore design variations addressing different types of change
- It is a design method in which the output – image, sound, architectural models, animation – is generated by a set of rules or an Algorithm, normally by using a computer program

Generative Design

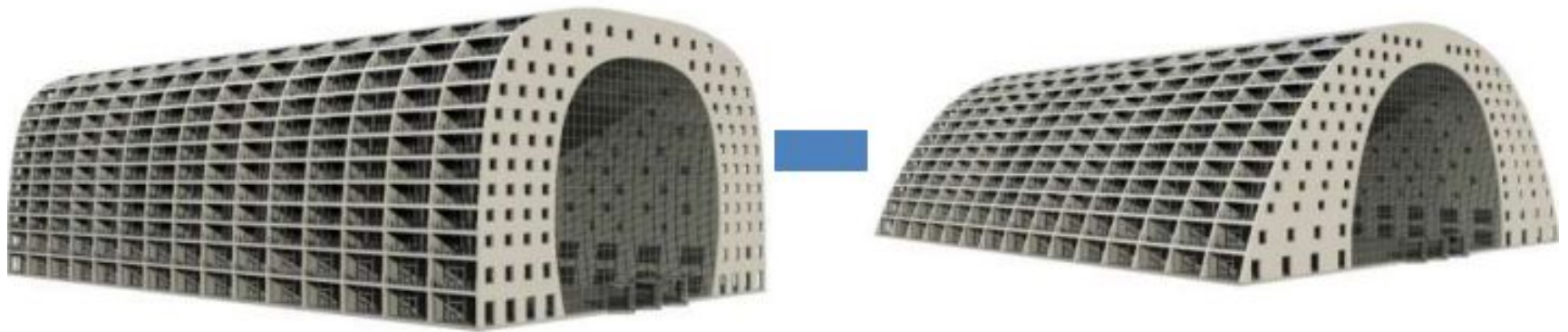
- Typically, generative design has:
 - A design schema
 - A means of creating variations
 - A means of selecting desirable outcomes
- It deals with the creation of images by using codes. An image is no more created manually but through a visual idea which is translated into a set of rules and then implemented in a programming language in the form of source code.
- The consequence is that such a program can not only create a single image but also completely re-design visual worlds by changing the parameters.

ADVANTAGES

- Produce unique designs and achieve variation of the same design.
- In the traditional use of CAD tools, when changes are needed, it requires too much time and effort to modify models & just one solution is represented and the exploration of different solutions requires manual changes to the model. But in GD, algorithms are translated in a programming language and defined parametrically, they automatically embrace change



CHANGE IN SHAPE OF
WINDOW



CHANGE IN SHAPE OF THE BUILDING



AFTER INTRODUCING SEVERAL CHANGES

DISADVANTAGES

- It lacks a clear theoretical framework in generative design.
- If manual changes are made to the generated model, these changes will be lost when the program is re executed and the model regenerated.
- It requires an obvious initial investment

Ideation, Sketching, Critiquing

- **Ideation:**
- Ideation is a process of generating many ideas in an open-minded environment.
- In the design thinking process, we ideate once we have completed discovery research and defined our problem statement, usually in the form of a how-might-we statement.
- It consists:
 - Methods
 - Brainstorming
 - Sketching
 - Storyboards

Ideation

- Coming up with ideas for design
- Starting point is divergent ... storming
 - Create many ideas
 - Explore very different possibilities
 - Fast and avoid critique
 - ***Collaborative***
- Later convergent ... critiquing
 - Compare what has been created
 - Evaluate it and identify best elements in each design
 - Potentially combine elements from different designs

Sketching

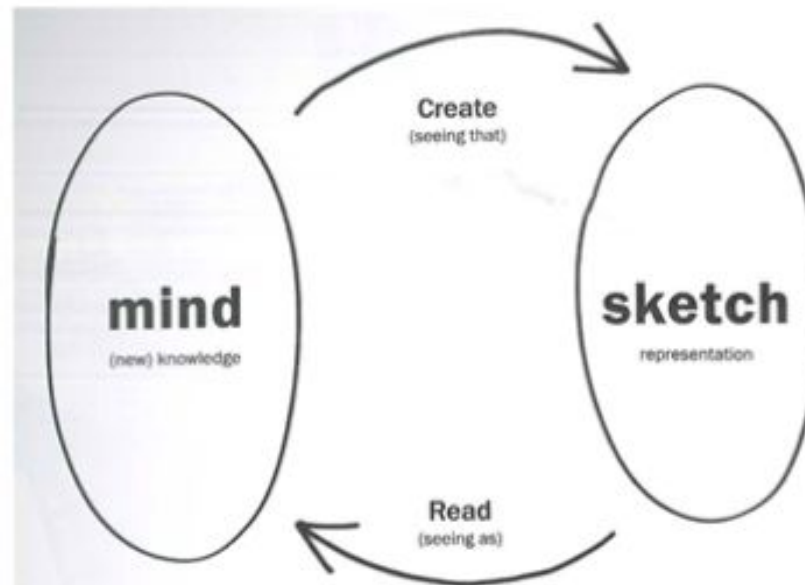


Figure 45: A Sketch of a Dialogue with a Sketch

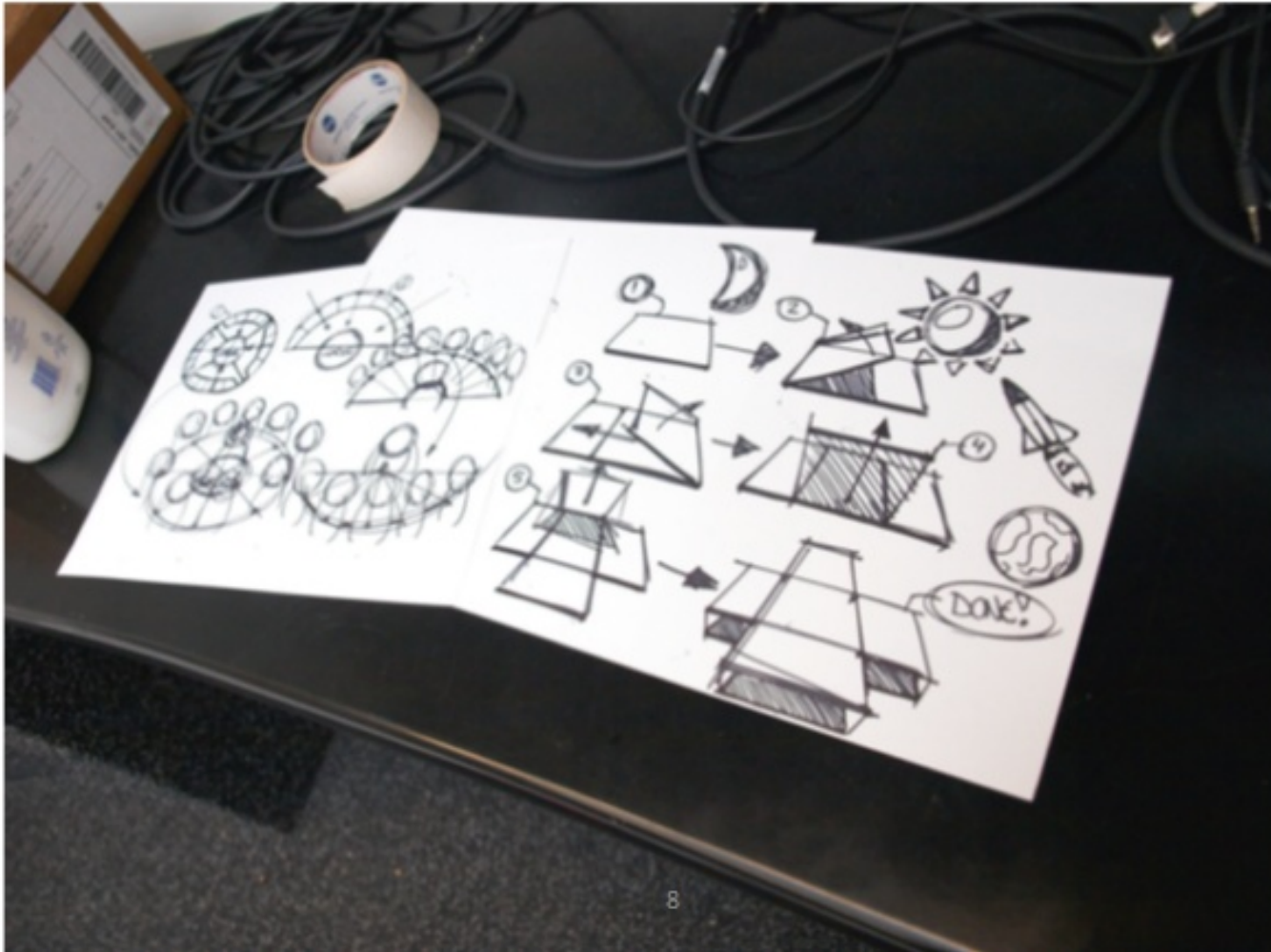
The "conversation" between the sketch (right bubble) and the mind (left bubble). A sketch is created from current knowledge (top arrow). Reading, or interpreting the resulting representation (bottom arrow), creates new knowledge. The creation results from what Goldschmidt calls "seeing that" reasoning, and the extraction of new knowledge results from what she calls "seeing as."

Embodied cognition – mind + drawing action

Sketching

- Sketches are not prototypes to refine a design
- Sketches are for exploring design ideas and expanding them
- Buxton: think of difference:
 - Sketch of a mobile phone design
 - Sketch of *the experience of using it*

Example, a sketch to think about a design



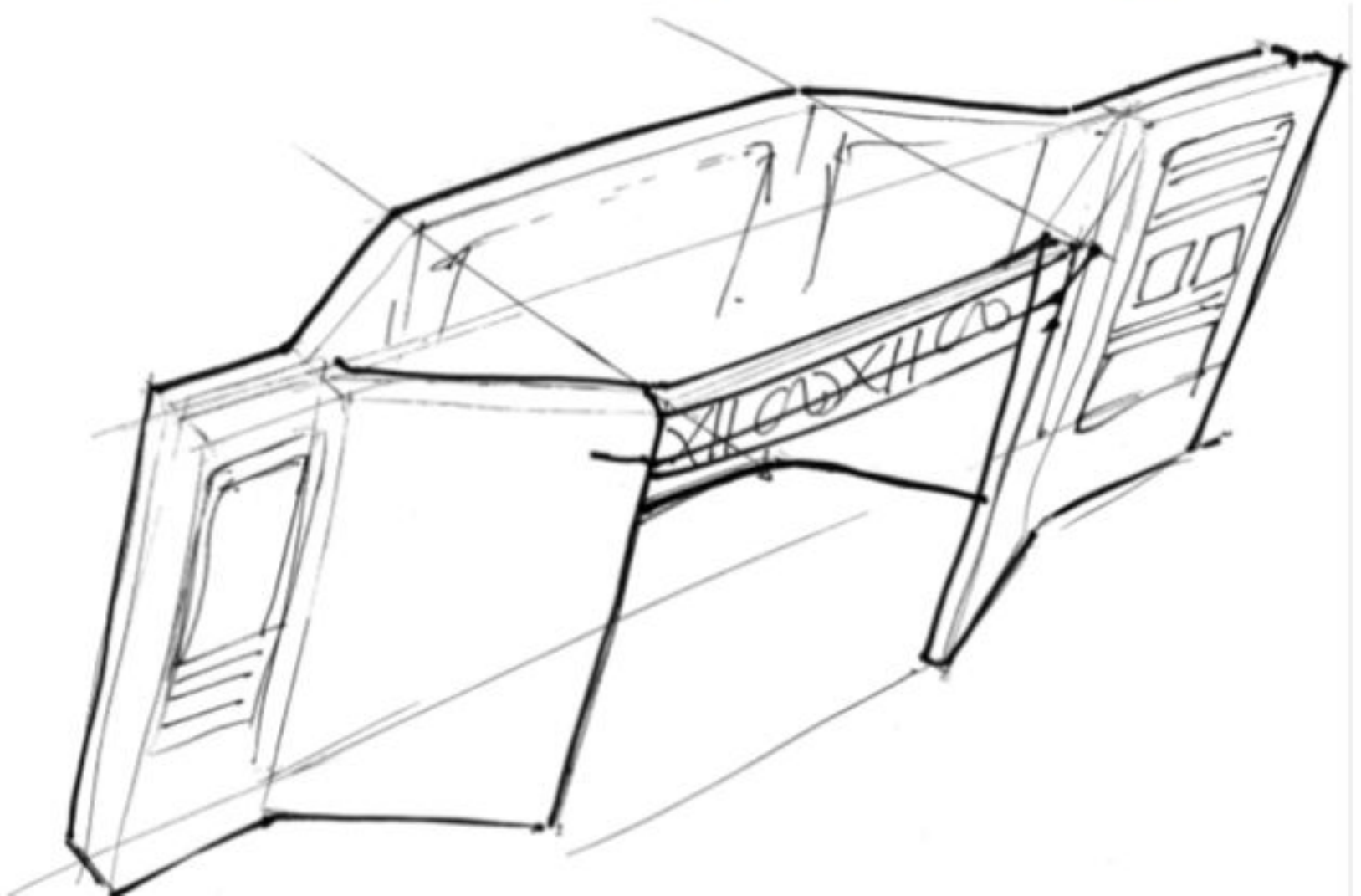
Sketching

- Sketching is essential to ideation and design
 - When you are designing, you must be sketching
- Adds “cognitive supercharging”

Use language of sketching

- The vocabulary of lines
 - Freehand “open” gestures
 - Roughed in, not connected precisely
 - Overlap, often extending a bit beyond corner
 - Sometimes they “miss” intersecting

Example, free-hand sketch of Ticket Kiosk System design



Use language of sketching

- Sketches are deliberately ambiguous
- Sketches are abstract, leaving “holes” for interpretation, imagination

Buxton's defining characteristics

- Everyone can sketch; you do not have to be artistic
- Most ideas are conveyed more effectively with a sketch than with words
- Sketches are quick and inexpensive to create; they do not inhibit early exploration
- Sketches are disposable; there is no real investment in sketch itself

Buxton's defining characteristics

- Sketches are timely
 - Can be made just-in-time
 - Done in-the-moment
- Sketches should be plentiful
 - Entertain large number of ideas
 - Make multiple sketches of each idea
- Textual annotations play essential support role

- Ideation sketching

- Everyone in turn, start throwing out ideas for discussion.
- Then, in parallel, each make sketches simultaneously (flowing the description of sketching in the book). Remember that a sketch is not just a picture; it's a conversation.
- No critiquing for 5 minutes – or while ideas flow.
- No idea is too far out.
- When that well starts running dry, switch to critiquing and evaluate the ideas, winnowing out the most promising ones.

Storyboards

- Sequence of visual “frames”
- Illustrating interplay between user and envisioned system
- Brings design to life in graphical “movie clips”
- Freeze-frame sketches of stories of how people will work with system.
- Visual design scenarios, envisioned interaction design solutions

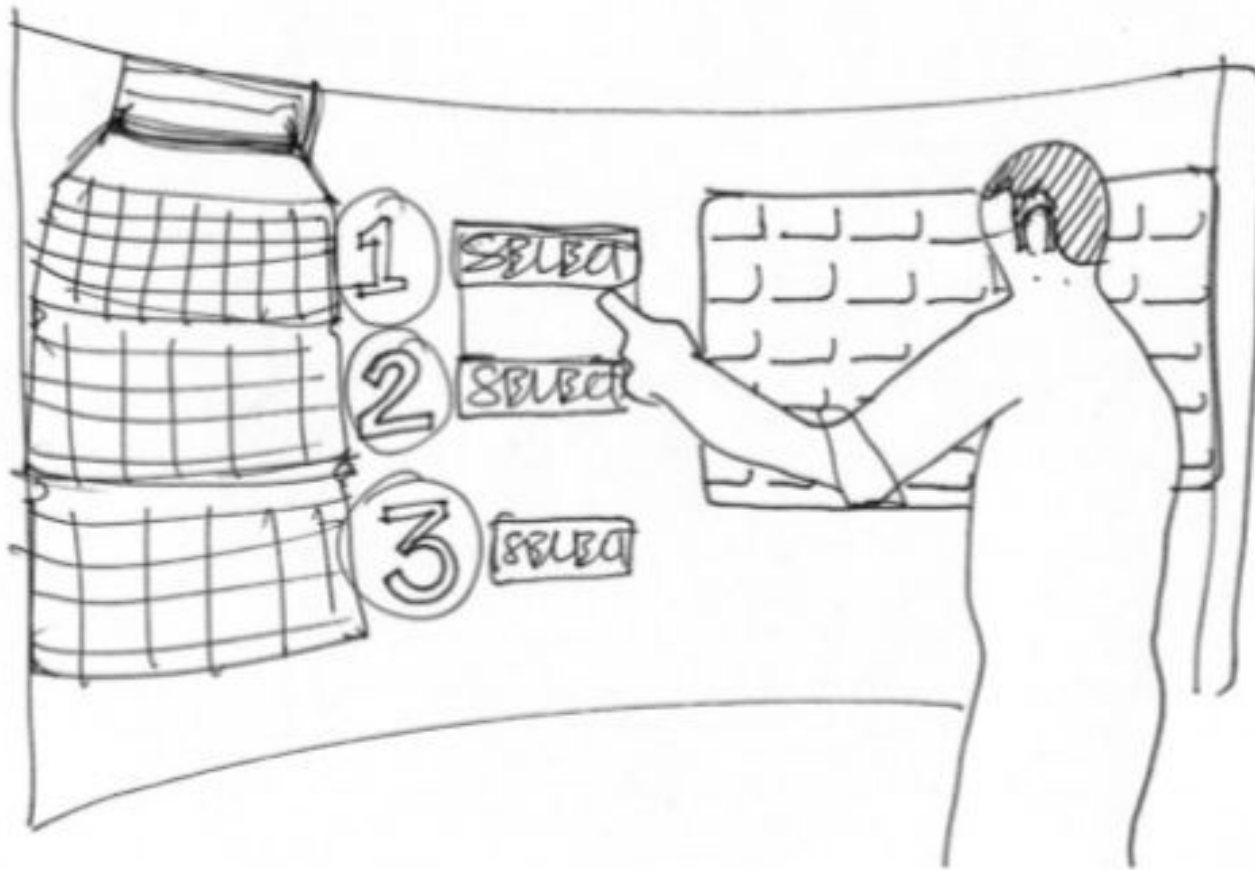
Include things like these in your storyboards

- Hand-sketched pictures annotated with a few words
- All work practice that is part of task, not just interaction with system
 - Example, include telephone conversations with agents outside system
- Sketches of devices and screens

Include things like these in your storyboards

- Any connections with system internals, for example, flow to and from a database
- Physical user actions
- Cognitive user actions in “thought balloons”
- Extra-system activities, such as talking with a friend about what ticket to buy

Example: Interaction perspective storyboard sketches



Example: Interaction perspective storyboard sketches



Example: Interaction perspective storyboard sketches





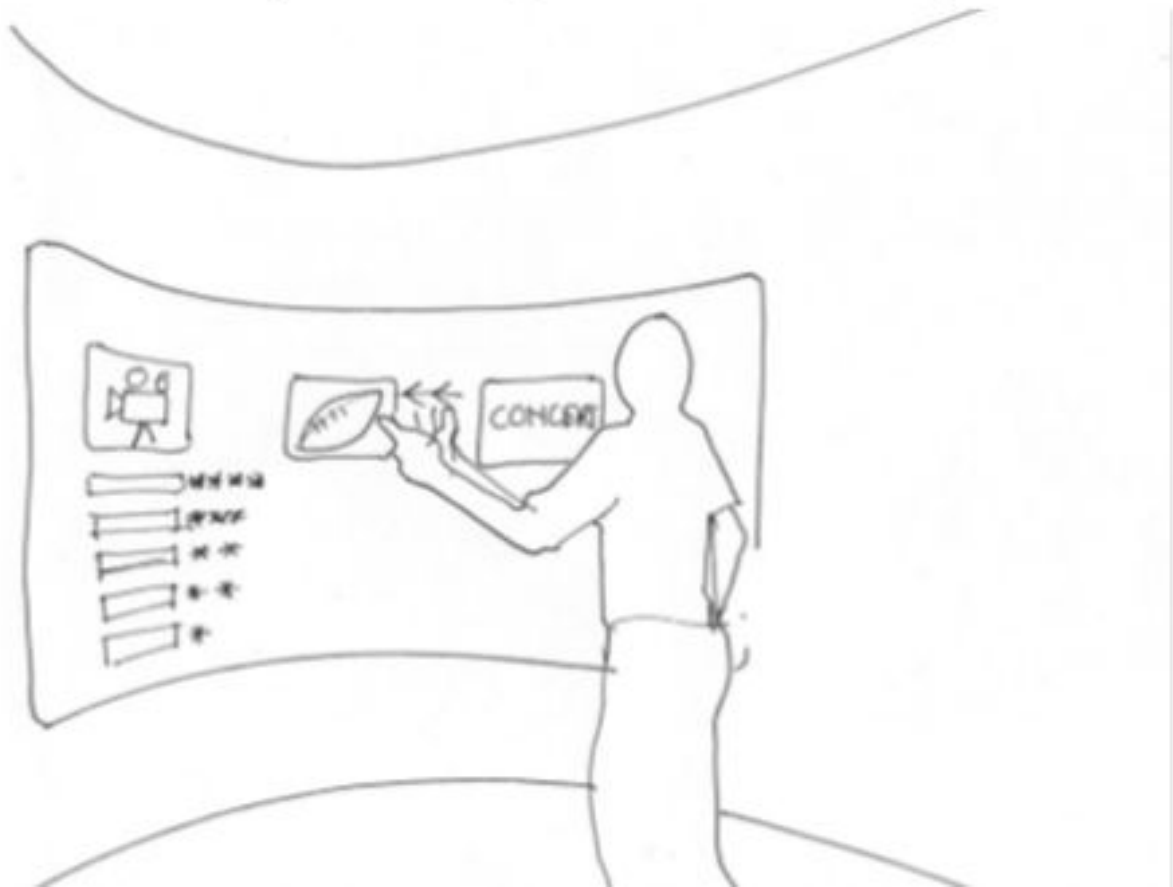
Phase 1: Paper Prototype



We sketched around for some time and picked certain view layouts, dismissing ugly and clumsy ones. That's an easy start, took us 15 minutes.

Example, storyboard transition frame

- Thought bubble explaining state change



Design Critique

- A design critique is **performed in order to evaluate a design and provide feedback on whether** or not it meets business objectives and user needs.
- Design critiques encourage collaboration within teams and provide a boost to positive team culture toward improving design ideas.

Design Critiques:

The right facilitation process can foster an efficient, honest feedback loop.

NN/g



Why Design Critique?

- to encourage a positive culture to improve products
 - to boost open exploration and discussion of designer's intent and the choices made to reach the designer's objectives
 - to detach emotionally from design by receiving feedback from other designers
 - to help designers refocus their work in areas that fall short while learning about their design process
- Design Critique - Feedback that is NOT use

3 underlying themes of effective critiques:

- **Clear scope for the conversation.** Too often critiques become unwieldy due to lack of scope. Critiques will only prove beneficial if there are unambiguous boundaries for what can and should be critiqued. Once boundaries are set, participants, duration, and formality can be determined.
- **Agreed-upon design objectives for the work.** In order to analyze a design and whether it meets its goals, there must be agreement on the problem that needs to be solved. This likely means a clear understanding of users and their needs. Without these, any feedback is subjective and baseless.
- **Conversation rather than command.** Commands, or directives, can very quickly ruin the exact purpose of the critique, which is to foster open discussion in order to improve the outcome.

There are two main facilitation approaches to UX critiques:

- **Round robin.** Participants share their perspectives one by one, making their way around the table. This method provides two clear advantages. First, everyone contributes. Second, the process feels democratic: you can start at a random place at the table, and anybody has a chance at going first (if not the first time, then the next time).
- **Quotas.** The facilitator gathers a specific, predetermined number of positive and negative comments from each participant. For example, each participant could share two aspects of the design that seems to accurately meet users' needs, and one aspect that could be improved. This approach specifically should be used only as a way to initiate conversation. Once there is a natural exchange, the critique could carry on based on where the conversation goes (assuming it falls within the set scope).

Critique Pitfalls

- Keeping a critique on track and effective is hard work. Below are bad habits that can negatively impact critiques:
- Not agreeing on personas or objectives beforehand
- Scheduling too long critique sessions
- Taking feedback personally
- Rushing to problem solve in the moment
- Talking only about the negatives