

Interactive prototyping in the early stages of the design process

User Interface Design seminar

Aadjan van der Helm



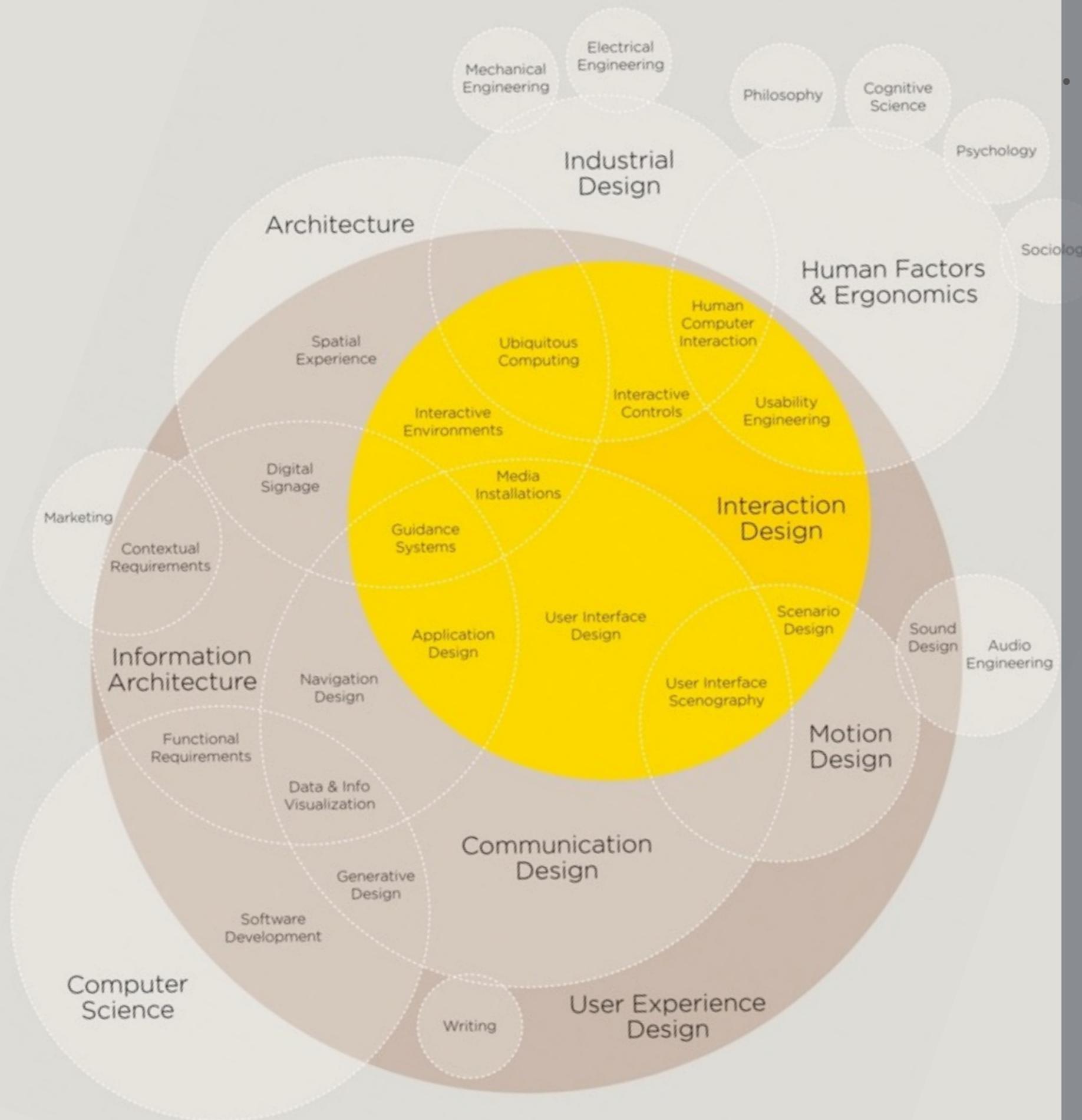
**Lecturer/researcher at
TUDelft**

**software engineering,
computer graphics,
computer networks**

interactive media art

interaction design

**Source: Concept Interaction style
Toolkit - Interactive Technology Design**



- **Design integrates knowledge from many disciplines**

- **Software engineers think: abstraction, maintenance, optimisation, efficiency, adaptability, code is poetry**

- **Designers think: concrete, one-off, towards experience, quick to evaluate**



**Mission:
Designing for
experience**

**Consider needs and abilities
of the user**

**Consider the environment
of use and the social
context**

**How does a user experience
a product**

**What is the understanding
of that product**

Source: Job Jansweijer - Konnekt



Products with embedded technology

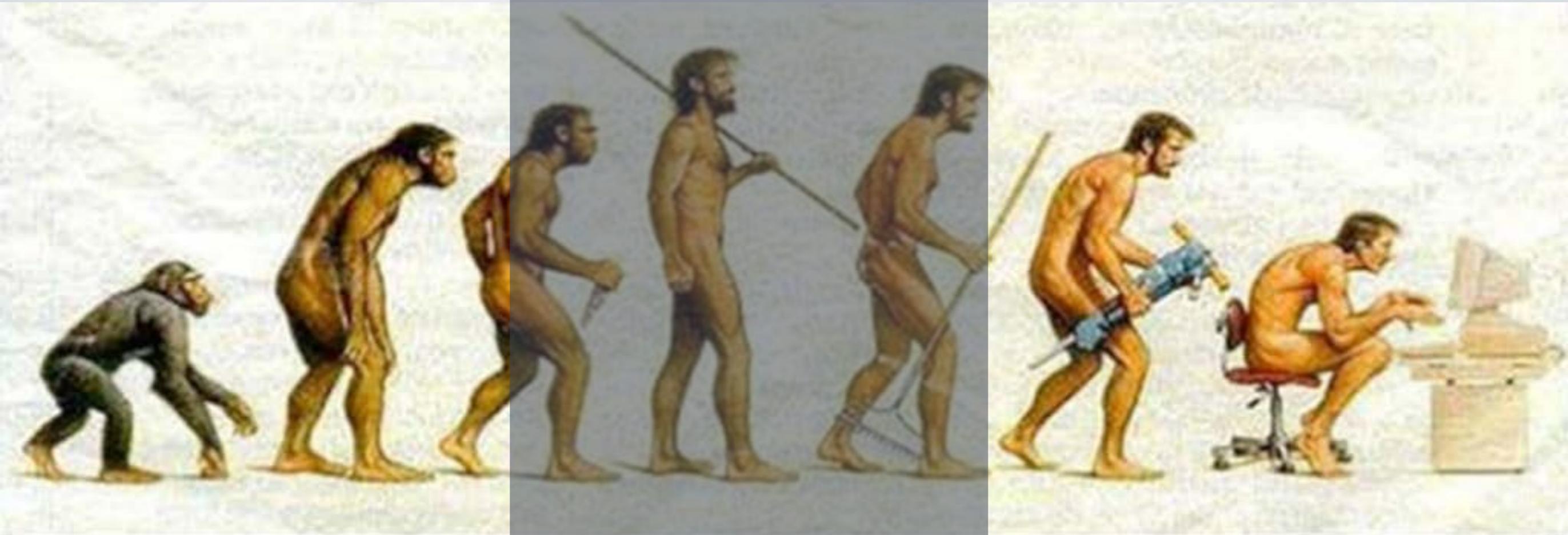
Sensors

Actuators

Microcontrollers

Evolution of man

From ape to knowledge worker



Homo WIMPi

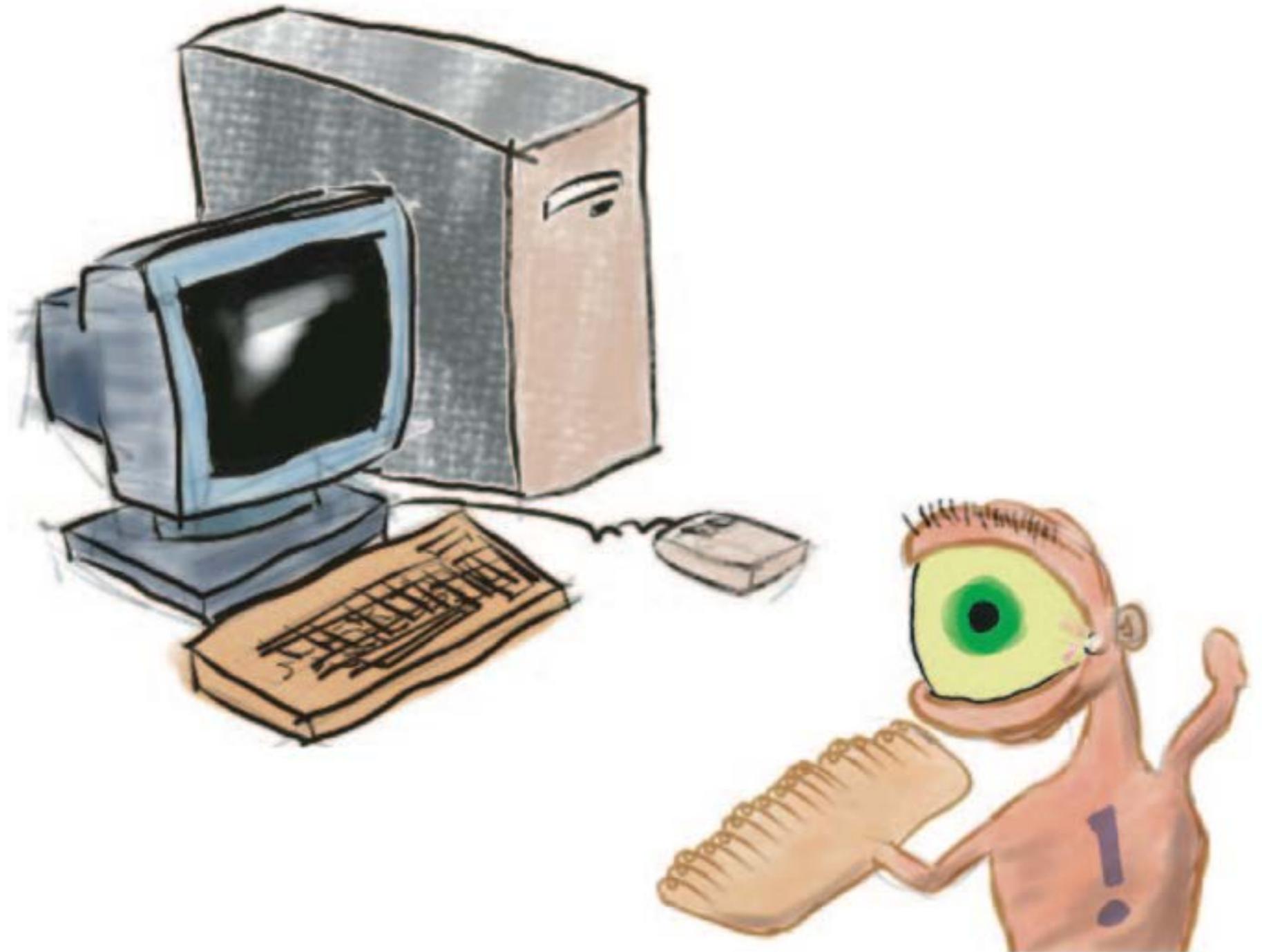
Windows

Interface

Mouse

Pointer

interaction





Our future?



Source: Wall-E - Pixar

Imagine the knowledge worker of the future as the artisan of today

Bodily skills of man are central to interaction

Involves multiple modes of understanding:

Visual, Aural, Tactile, Kinaesthetic, Spatial

**Bret Victor - UIST 2014:
On Humane Representation of Thought**



How to design for interaction



Source: Concept Traho -
Interactive Technology Design

INTERACTION DESIGN



Ingredients of
interaction design

What does the user

know

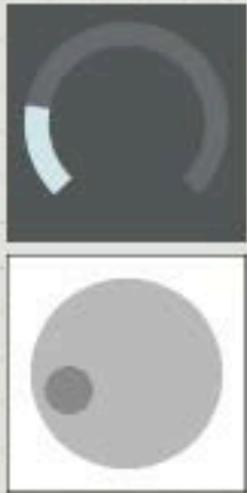
do

feel

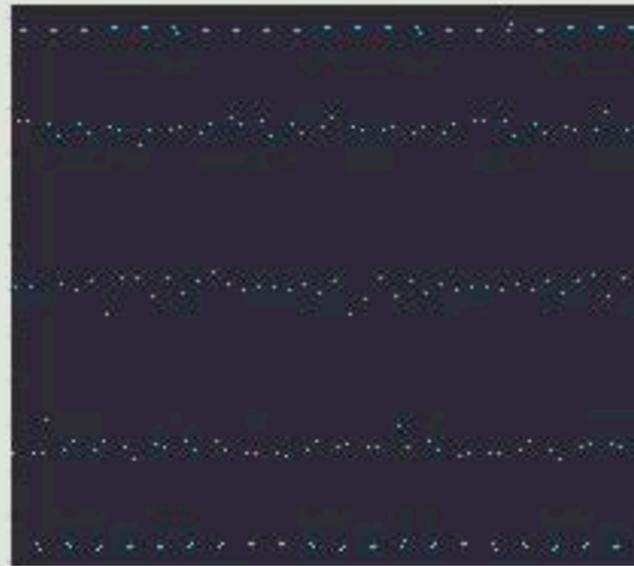
Ingredients of interaction design

Interactive behaviour of product
Feedback

Controls

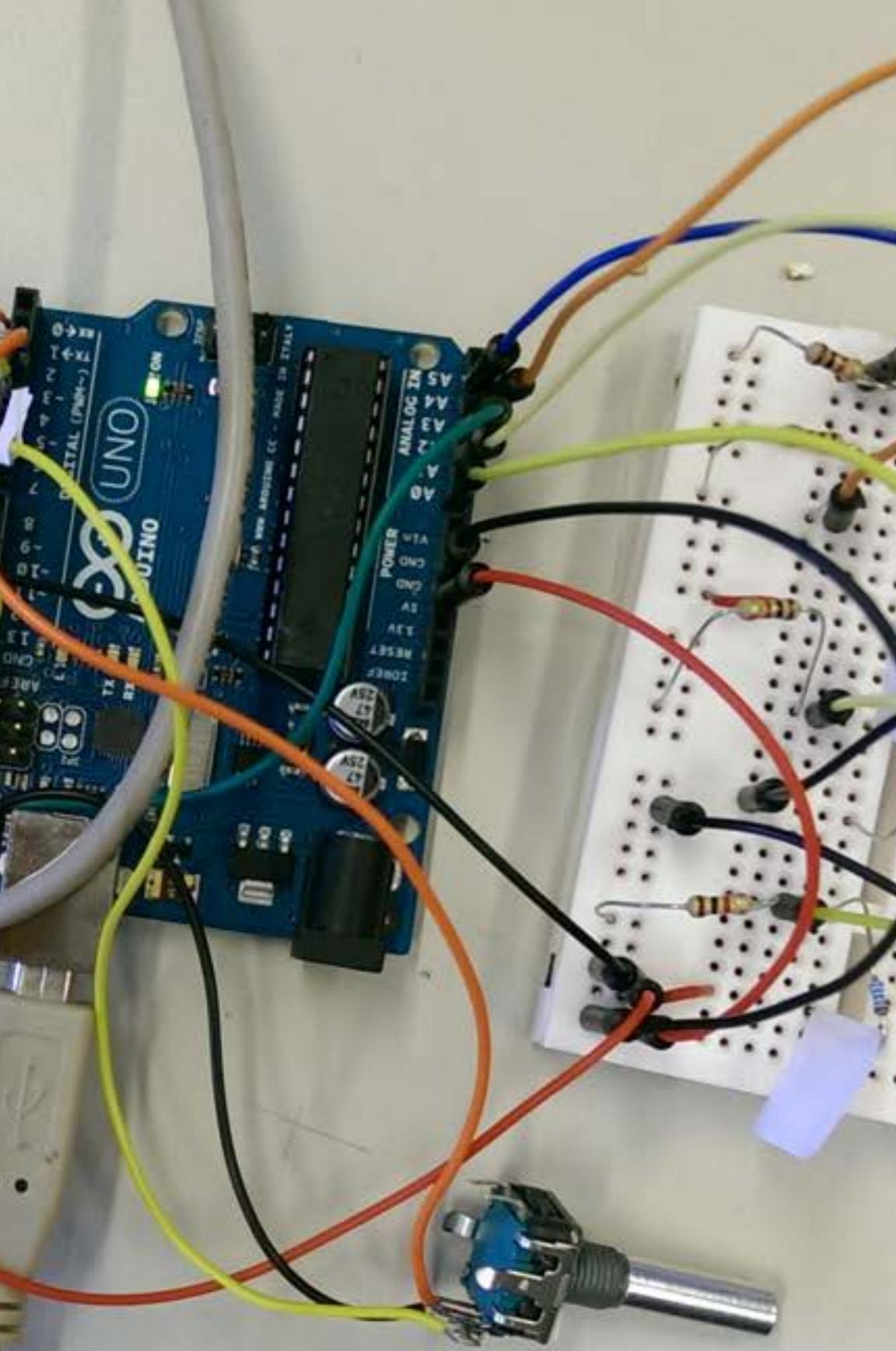


Mapping



Feedback





```
// These constants won't change:  
const int analogPin = A0;    // pin that the sensor is attached to  
const int ledPin = 3;       // pin that the LED is attached to  
const int threshold = 400;   // an arbitrary threshold level that's  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize serial communications:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // read the value of the potentiometer:  
  int analogValue = analogRead(analogPin);  
  
  // if the analog value is high enough, turn on the LED:  
  if (analogValue > threshold) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
  
  // print the analog value:  
  Serial.println(analogValue);  
  delay(1);          // delay in between reads for stability  
}
```

Ingredients of interaction design

electronics and
programming

Ingredients of interaction design

Multi disciplinary design teams



Donna Stam

Communicator/Reporter
(Concepter)



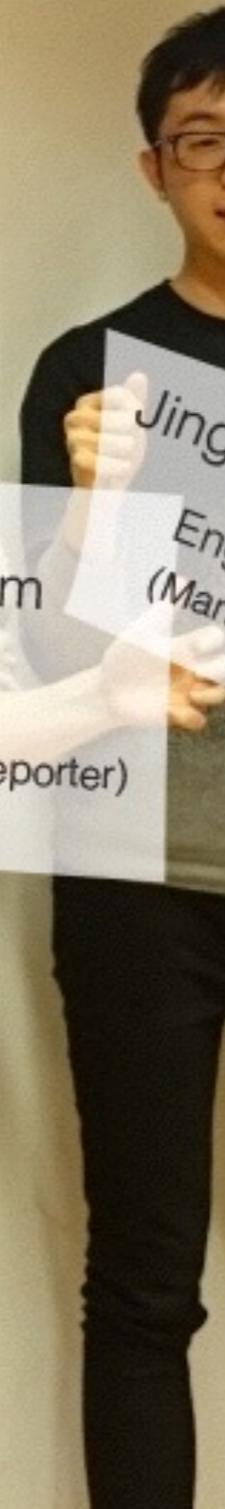
Jeppe Bijker

Builder
(Engineer)



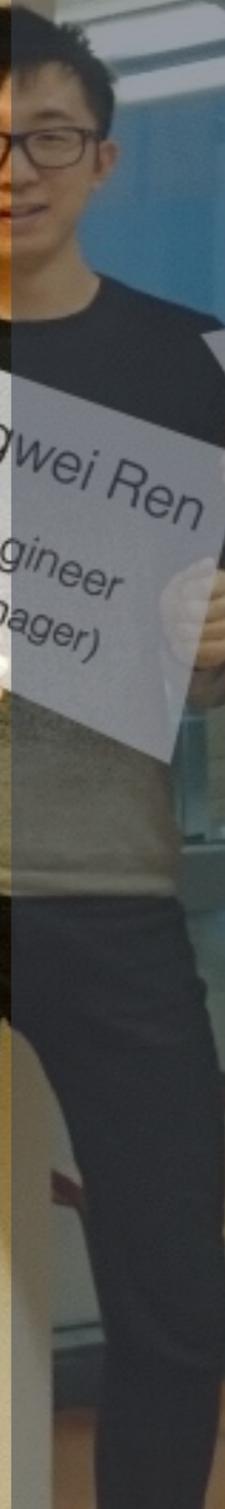
Rianne Ham

Manager
(Communicator/Reporter)



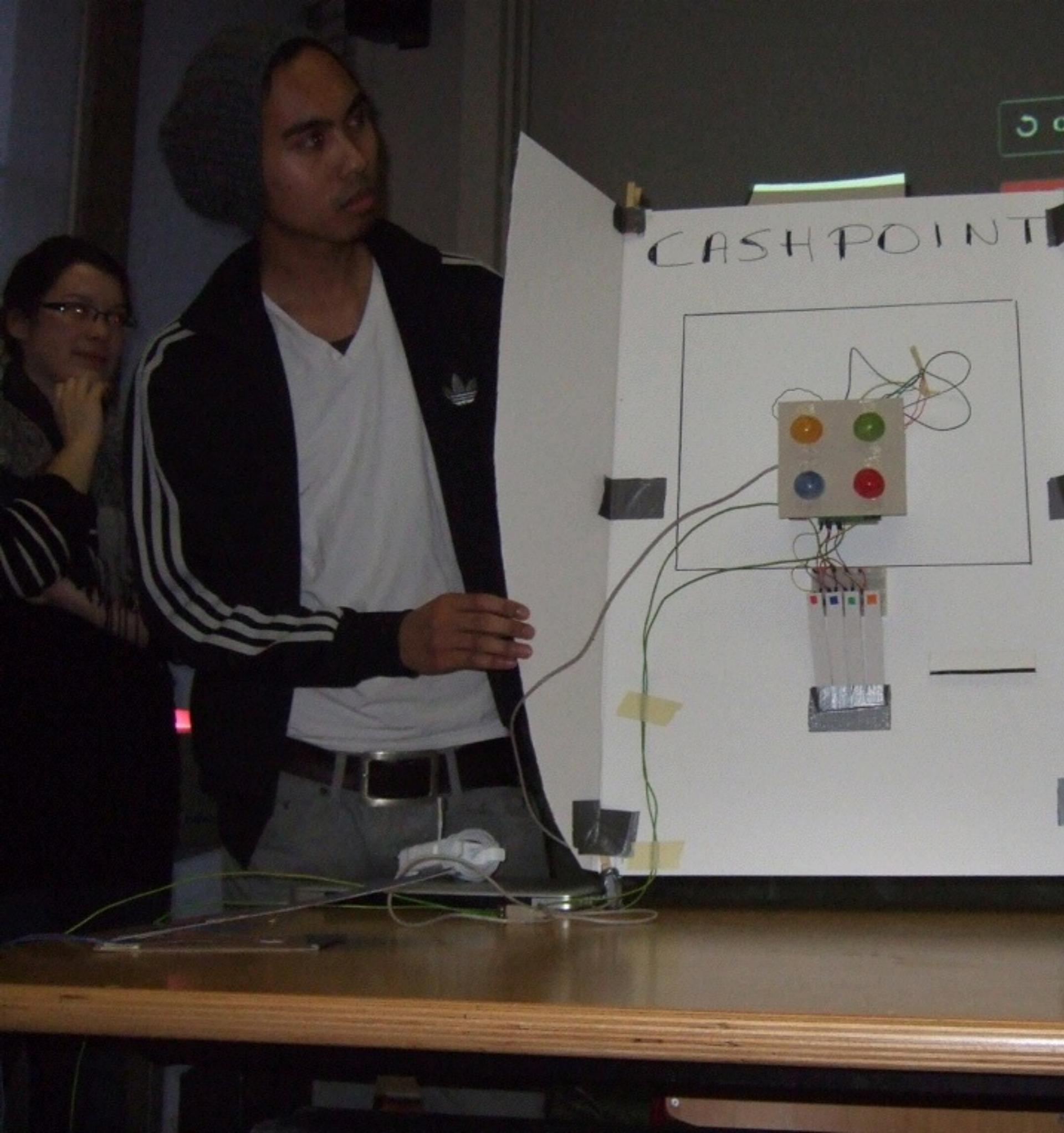
Jingwei Ren

Engineer
(Manager)



Kersti Rimmeld

Concepter
(Builder)



**Ingredients of
interaction design**

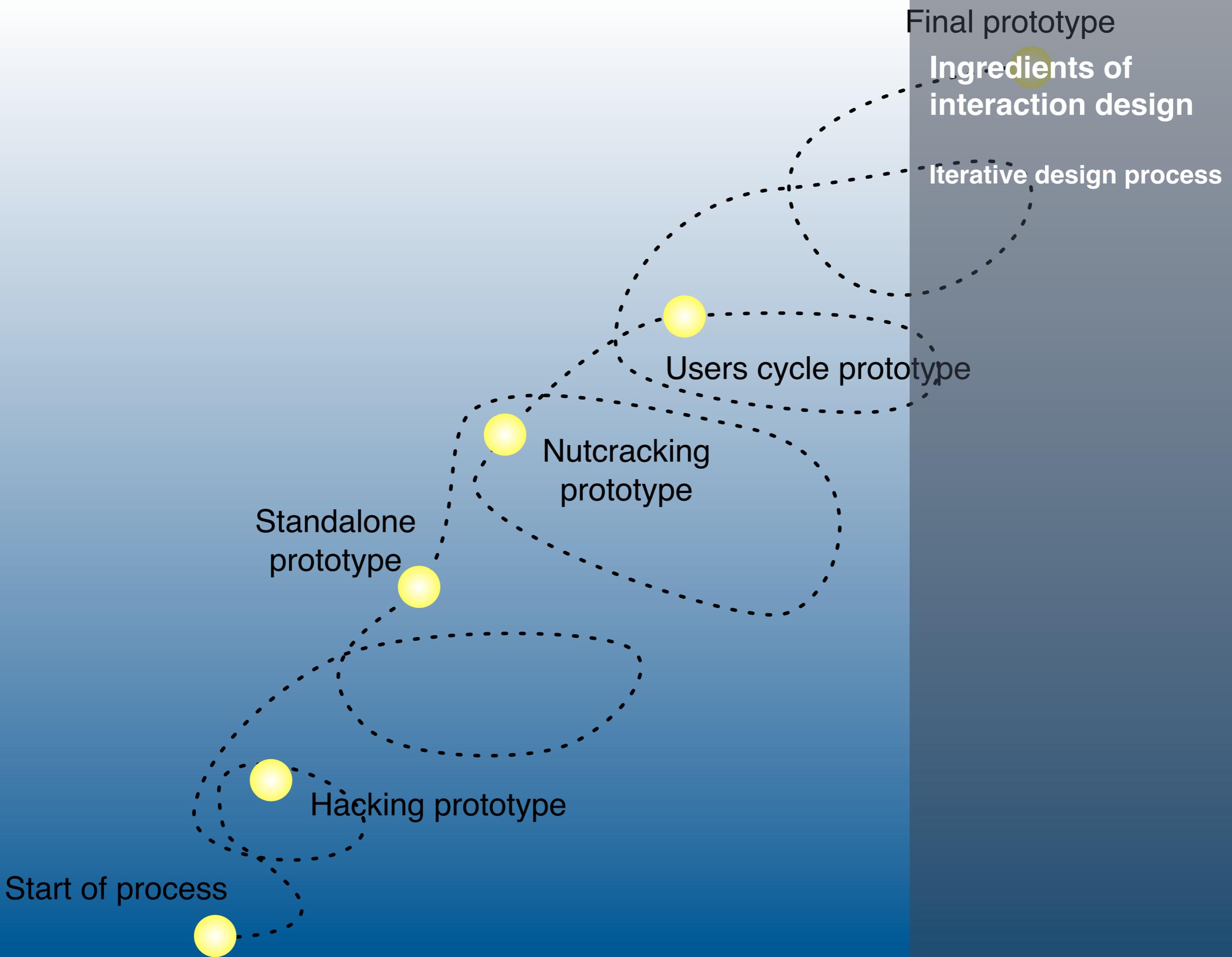
Prototypes

Making-first as a strategy

Thinking, talking, inspiration

Faking, Wizard of OZ

**Sketchy first, progressive
refinement**



How to support the design process for interaction design





Tools

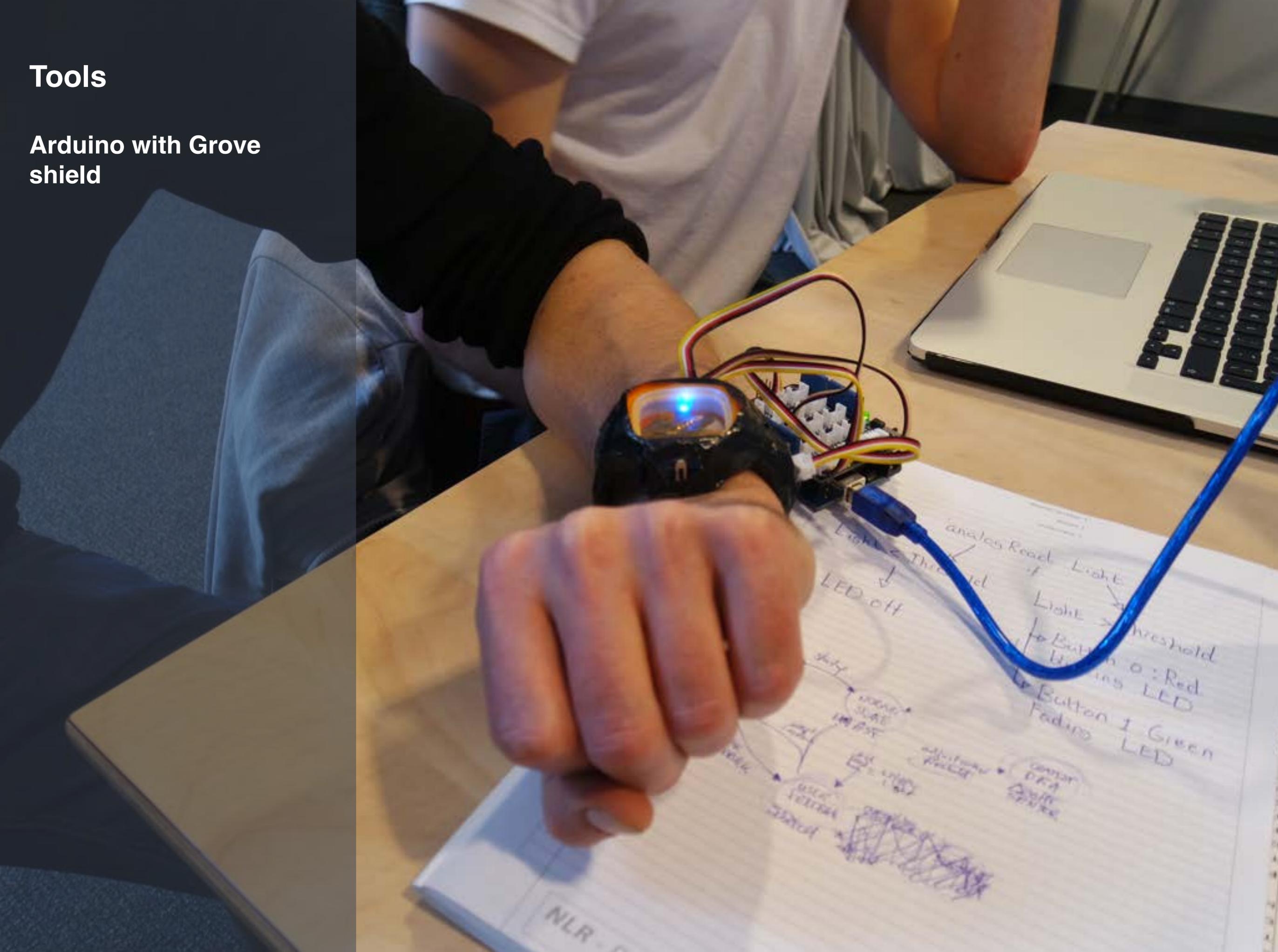
Random stuff

Tinkering material



Tools

Arduino with Grove shield



Tools

From interaction scenario to state transition diagram

Machino: an Arduino IDE for programming behaviour through state machines

The screenshot displays the Machino IDE interface. On the left, a state transition diagram for a lamp is shown with five states: 'Auto (on)', 'Auto (off)', 'Always off', 'Always on', and 'Dimmable'. Transitions are labeled with events and actions, such as 'light / turn off led' and 'button press / turn on led'. The 'Always off' state is a double circle, indicating it is the initial state. The 'Dimmable' state is a yellow circle. On the right, the code editor shows the implementation of these states. The code includes a `buttonPress()` function that checks for a button press and updates the `buttonWasDown` variable. It also includes `dark()` and `light()` functions that check the light sensor value. The `turnOffLed()` and `turnOnLed()` functions are also shown. Below the code editor, the 'Output' tab shows the binary sketch size: 3.682 bytes (of a 32.256 byte maximum) - 11% used.

```
boolean buttonWasDown = false;

// -----
// -- Setup & Loop --
// -----

void setup() {
}

void loop() {
  updateStateMachines();
}

// -----
// -- Conditions --
// -----

boolean buttonPress() {
  boolean buttonIsDown = button.get();
  boolean justReleased = (buttonIsDown == false && !buttonWasDown);
  buttonWasDown = buttonIsDown;
  return justReleased;
}

boolean dark() {
  int value = lightSensor.get();
  return (value < 600);
}

boolean light() {
  int value = lightSensor.get();
  return (value > 600);
}

// -----
// -- Actions --
// -----

void turnOffLed() {
  led.off();
}

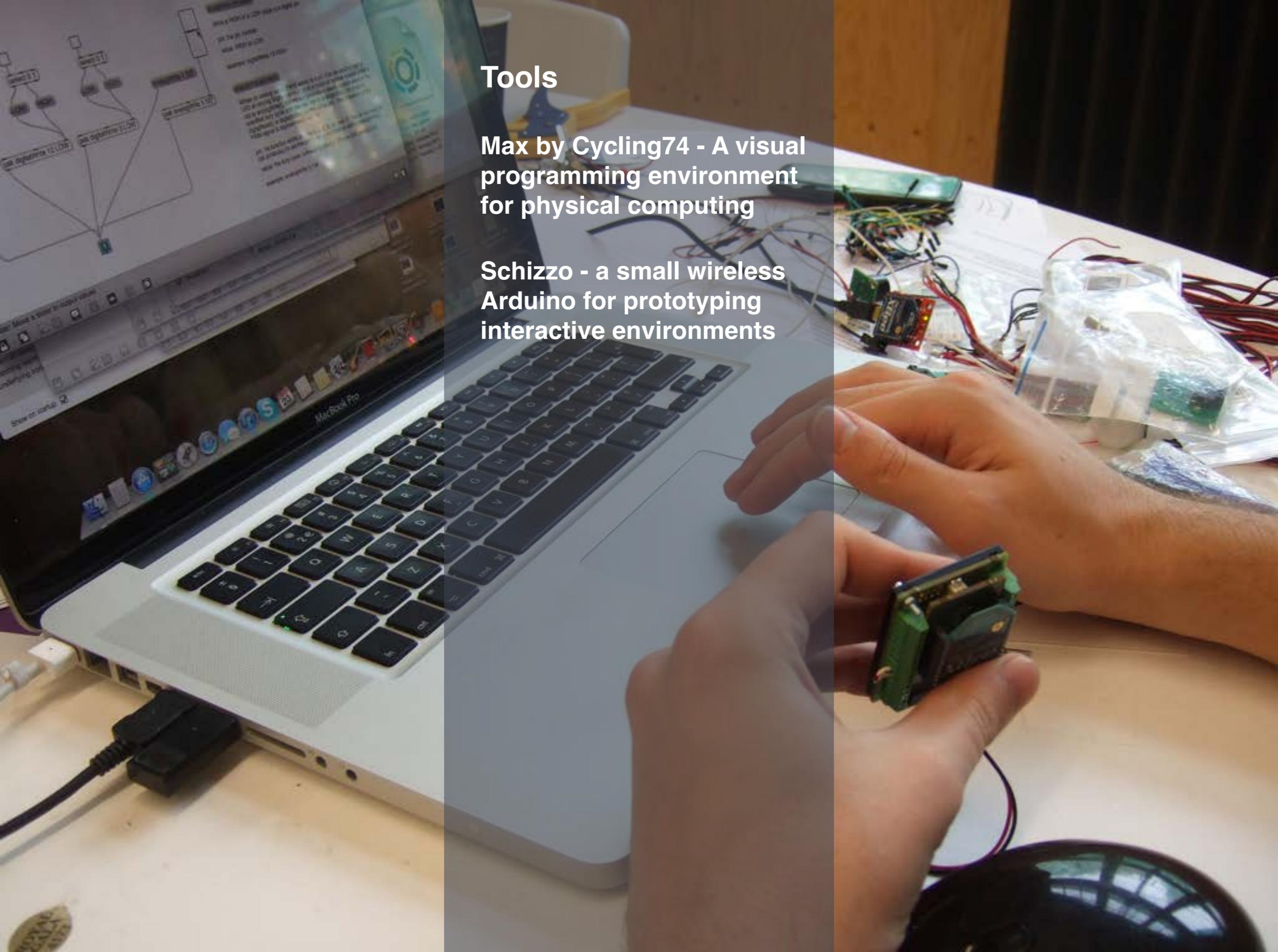
void turnOnLed() {
}
```

The Pattern Library window shows the 'buttonPressed()' pattern. It includes an 'Example applications' section with two bullet points: 'Detecting individual button presses.' and 'Detecting a sensor being released.' The 'Usage' section explains that the condition below checks if the button was just released, i.e. it completed one full press. The 'Variable' section shows the code to add to the variables section: `boolean buttonWasDown = false;`. The 'Condition' section shows the code to use for a condition: `boolean buttonPressed() { boolean buttonIsDown = ; // TODO: Get whether button is down boolean justReleased = (buttonIsDown == false && buttonWasDown); buttonWasDown = buttonIsDown; return justReleased; }`

Tools

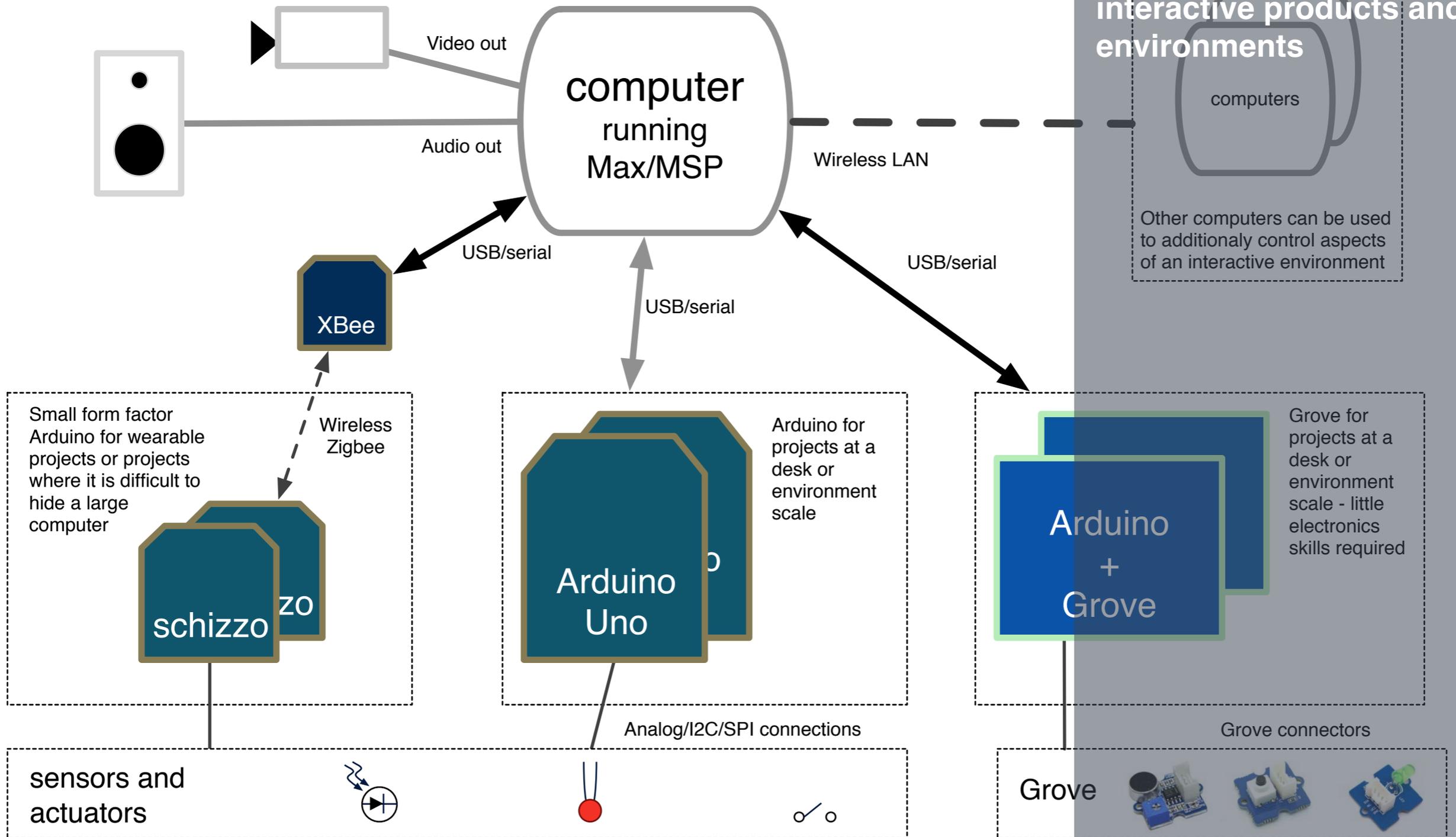
Max by Cycling74 - A visual programming environment for physical computing

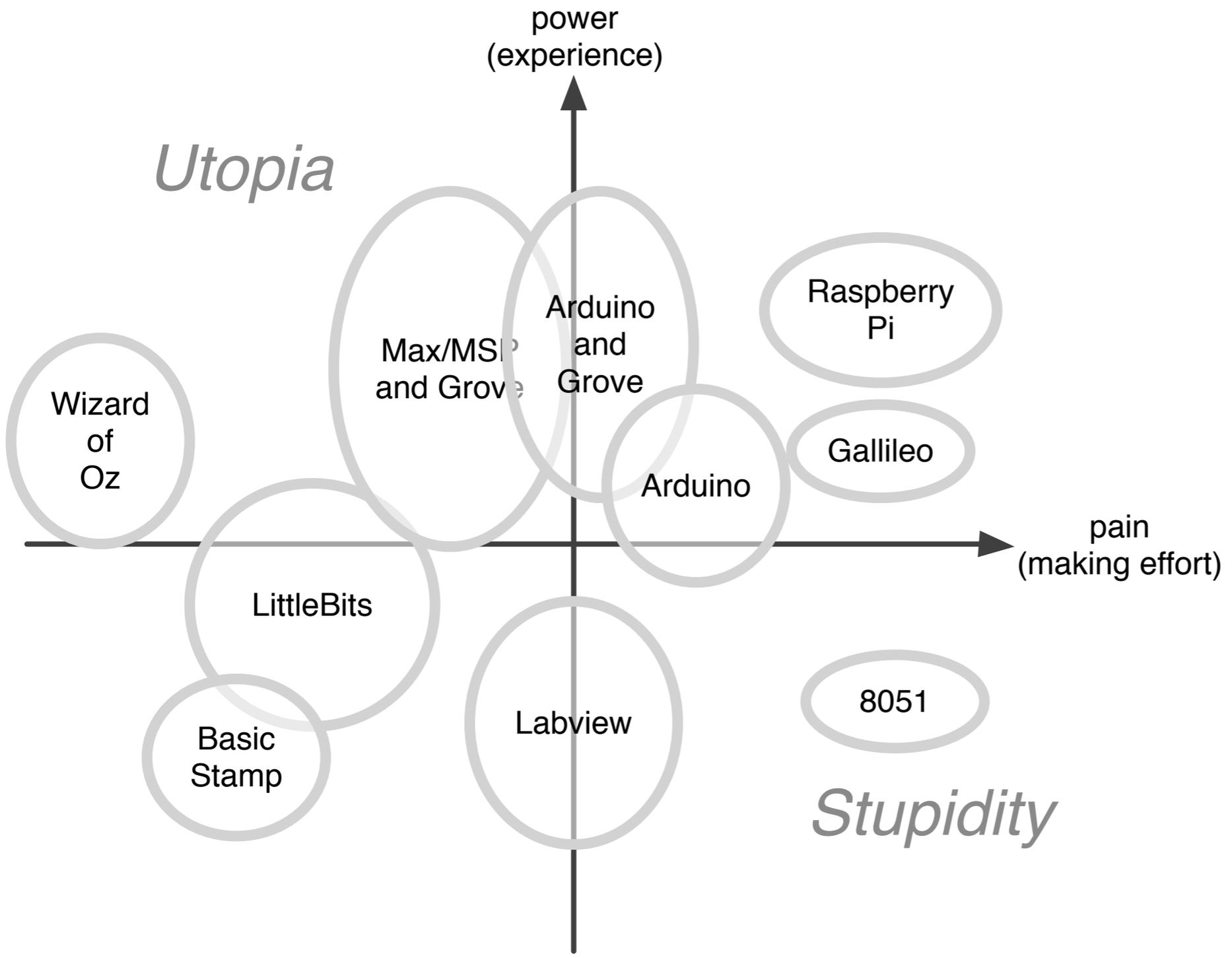
Schizzo - a small wireless Arduino for prototyping interactive environments



Technology in ITD

Reference diagram for hardware and software tools





Criteria selecting a technology tool for interaction design

Should support sketching

Large user community

Help/Examples/Tutorials

Rich capabilities: networking, video, audio, 3D, etc.



The Future

Technology tools to support the design of objects and environments to allow for embodied interaction