# Building Interactive Systems

# Course Introduction

## Professor Bilge Mutlu | Spring 2023

# About Me

- → Professor of Computer Science
- → *PhD*, Carnegie Mellon University, HCII, 2009
- → *Director*, People & Robots Laboratory

## How about you?

- → Introductions

# What is this course about?

A better question: *What is a "system?"*

...

...

And a follow-up question: *What is an " interactive system?"*

...

...

# What is this course about?

A better question: *What is a "system?"*

*"a set of connected things or devices that operate together"*
*"a set of computer equipment and programs used together for a particular purpose[1]"*

And a follow-up question: *What is an "interactive system?"*

*"Interactive systems are computer systems characterized by significant amounts of interaction between humans and the computer.[2]"*

[1] https://dictionary.cambridge.org/dictionary/english/system

[2] https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/interactive-systems

So, we're going to learn about :

*"a set of computer equipment and programs used together for a particular purpose... characterized by significant amounts of interaction [with] humans"*

Yes, but what of it?

## Learning Goals (high-level)

1. **Analyze:** Define and identify interaction systems, including archetypes, components
2. **Design:** Devise, prototype, evaluate new interactive systems
3. **Share:** Make and communicate "systems" contributions to literature

# What is an interactive system?

For a computer artifact to qualify as an "interactive system," it has to satisfy:

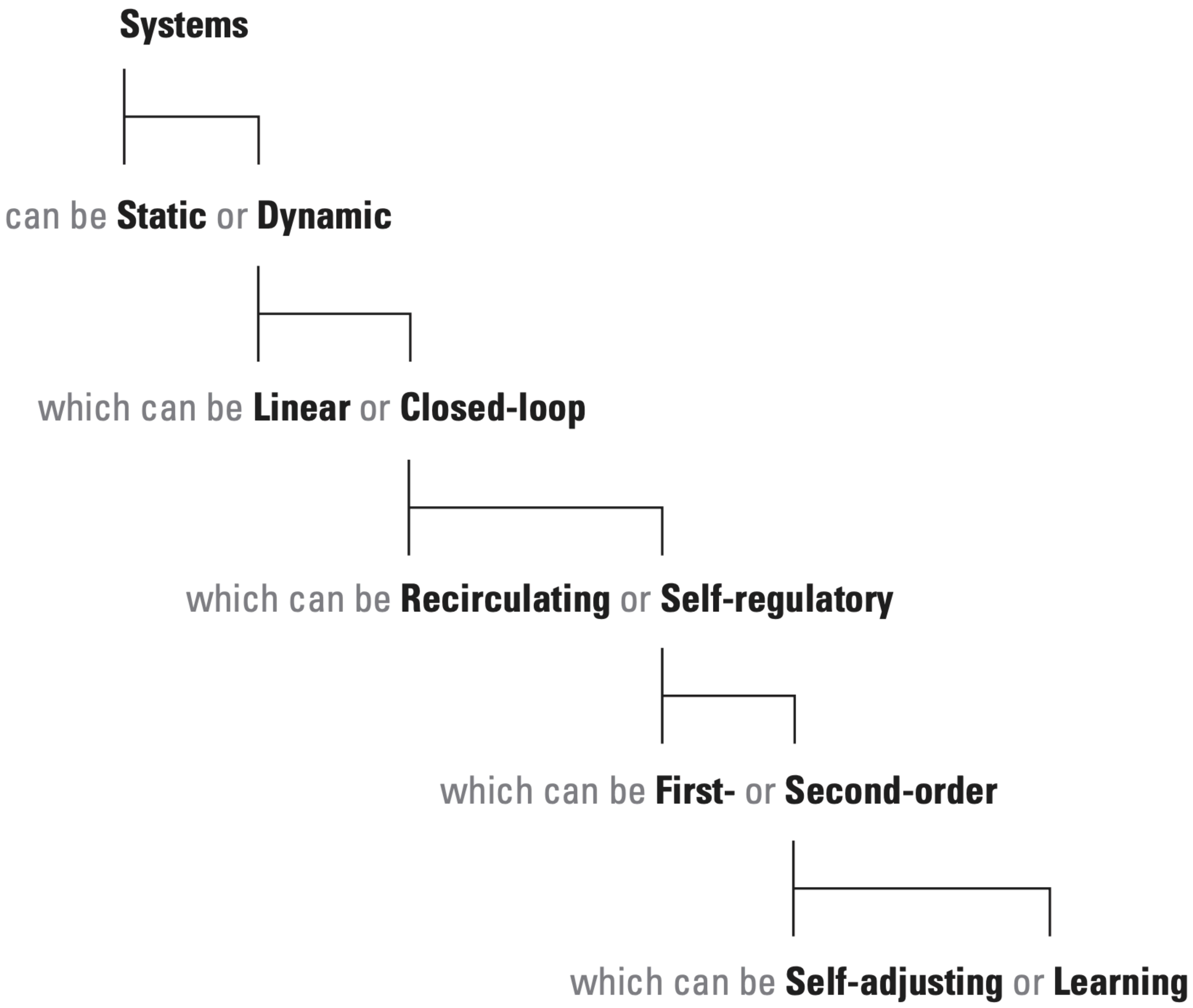
1. **System requirement:** made up of a set of equipment and programs used together for a particular purpose
2. **Interactivity requirement:** involving significant interaction between humans and the computer

# System Requirement[3]

| | **Object**<br>Component | **System**<br>Systems of components | **Ecosystem**<br>Systems of systems |
|---|---|---|---|
| **Why?**<br>Pragmatic | Communicate course information | Manage course activities | Learn course content |
| **What?**<br>Semantic | Webpage | Website | Class |
| **How?**<br>Syntactic | Markdown | Model, navigation, materials | Website, lectures, assignments, project |

[3] Dubberly. 2010. *The Space of Design*

# Types of Systems[4]



Systems

can be **Static** or **Dynamic**

which can be **Linear** or **Closed-loop**

which can be **Recirculating** or **Self-regulatory**

which can be **First-** or **Second-order**

which can be **Self-adjusting** or **Learning**

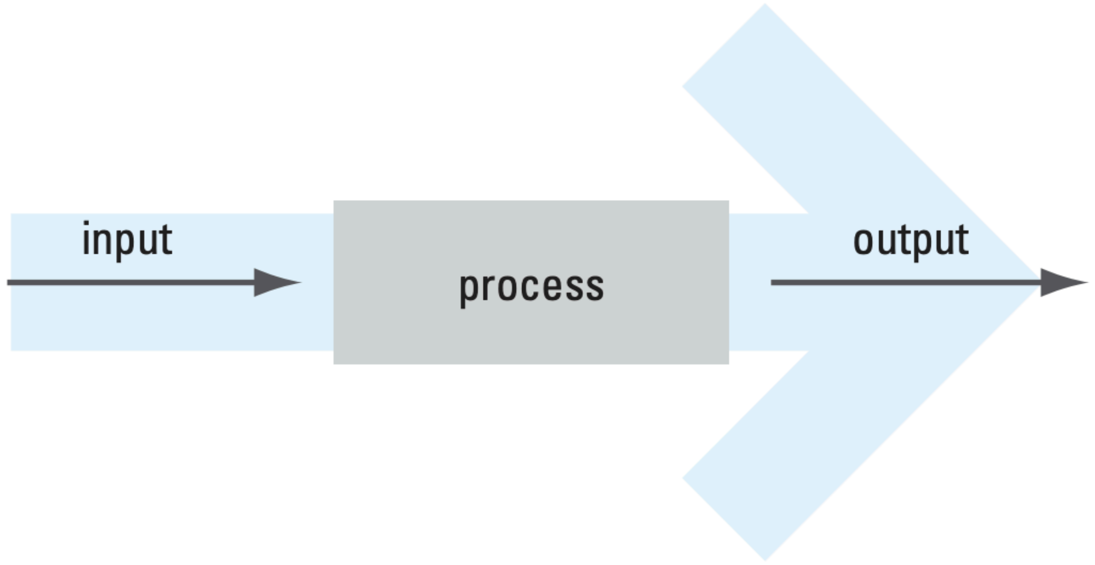
[4] Dubberly et al. 2009. What is interaction? Are there different types?

# A Linear System[4]

A system that responds to input by engaging in a process that produces an output.

**Example:** light

- → **Input:** flip of a switch
- → **Process:** electrical conductance in circuitry
- → **Output:** light emitted



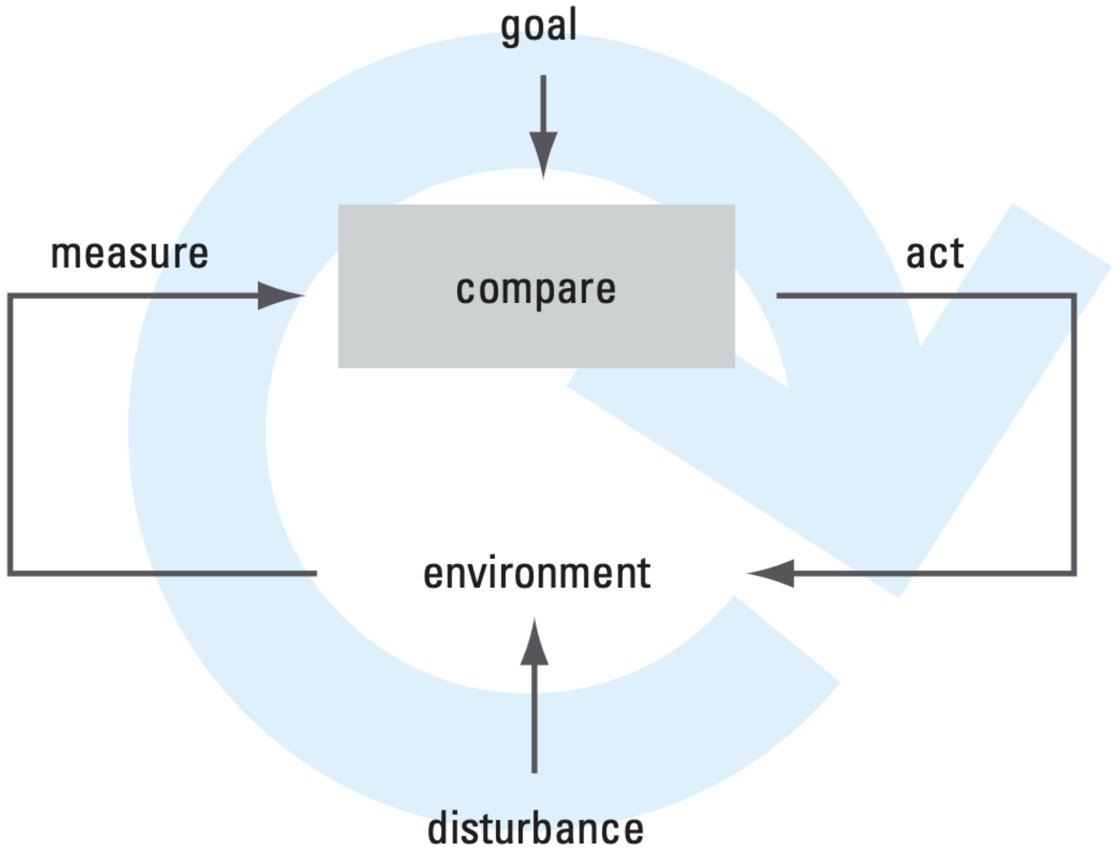
[4] Dubberly et al. 2009. What is interaction? Are there different types?

# A Self-Regulating System[4]

A system that acts based on a goal, measures its effects in the environment, and regulates its actions.

**Example:** elevator

- → **Goal:** travel to preset floors
- → **Act:** travel a set distance
- → **Measure:** sense location
- → **Compare:** update action



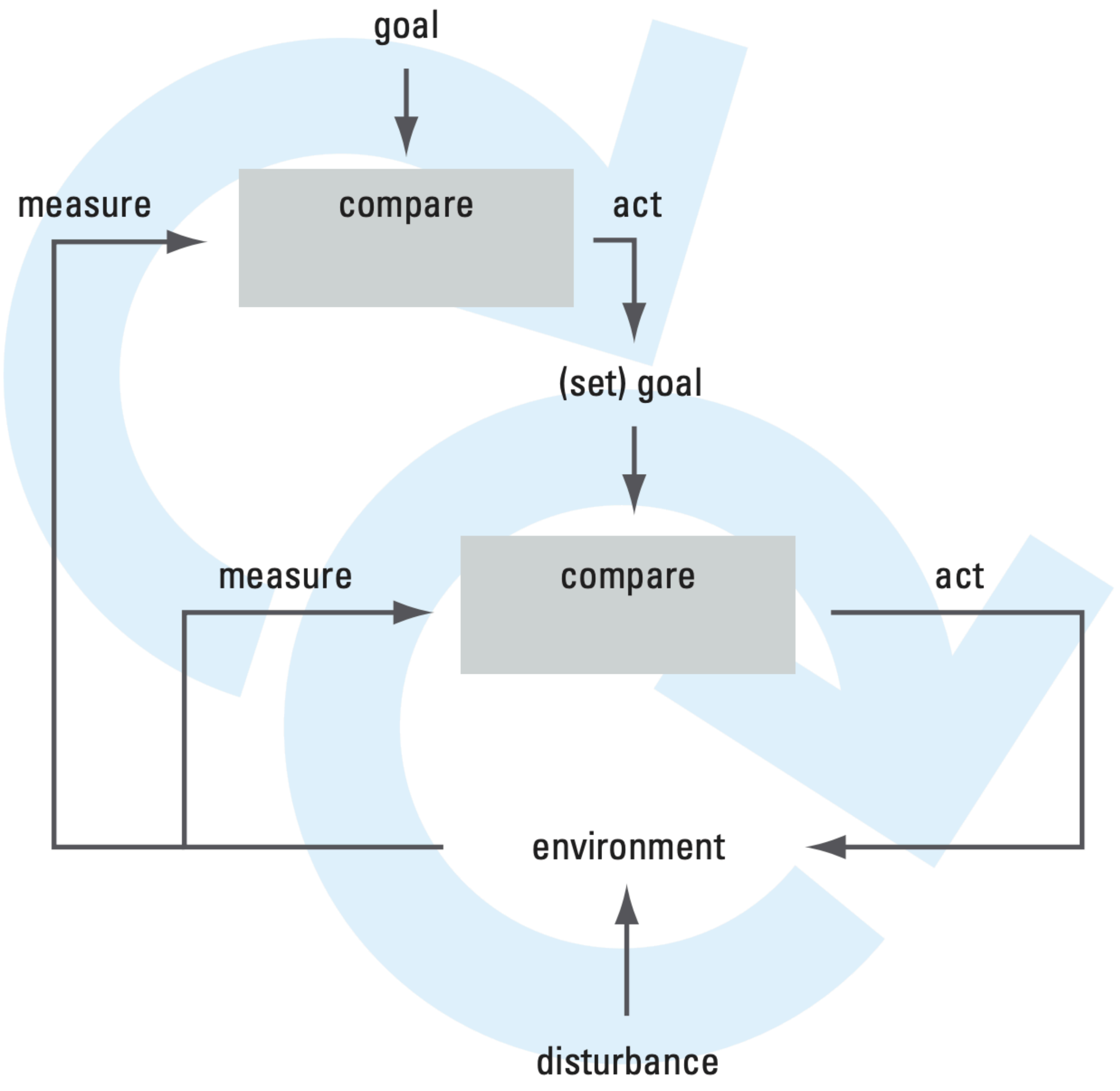
[4] Dubberly et al. 2009. What is interaction? Are there different types?

# A Learning System[4]

A self-regulation system where the goal of the system is learned from sensor information.

**Example:** smart thermostat

- Self-regulating system:
  - Goal: set by user
  - Act: turn on furnace
  - Measure: sense temperature
  - Compare: update action
- Learning system:
  - Goal: learn user patterns
  - Act: adjust goal based on learned pattern
  - Measure: sense variability
  - Compare: update action



[4] Dubberly et al. 2009. What is interaction? Are there different types?

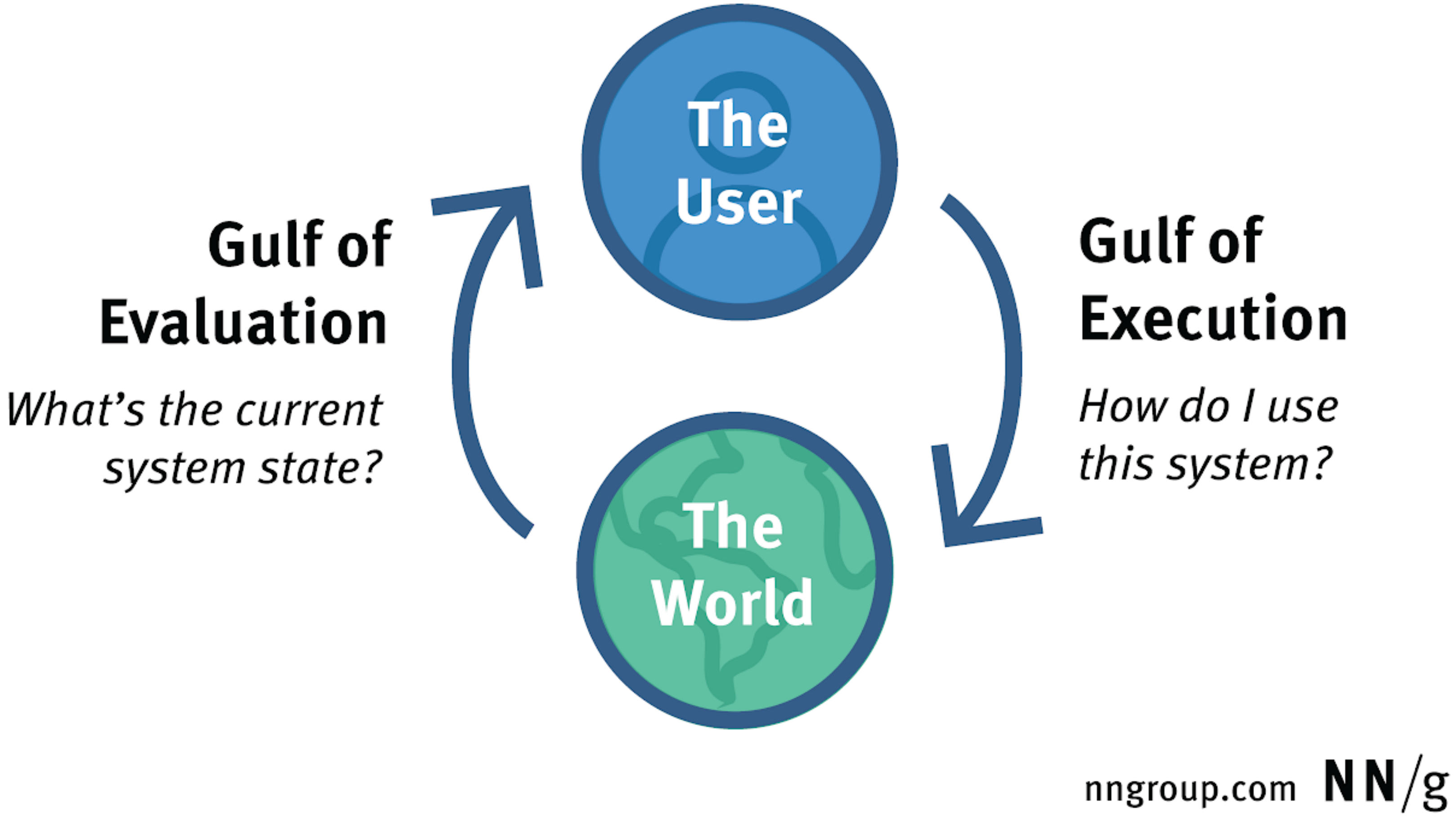
# Interactivity Requirement[5]

Two "gulfs" that users must overcome when they interact with systems:

1. Gulf of **evaluation** — "What is the system status?"
2. Gulf of **execution** — "How do you I use the system?"

Interactive systems similarly have to:

1. Accept **input** from users — "What is the user doing?"
2. **Display** system status — "How do I inform the user?"

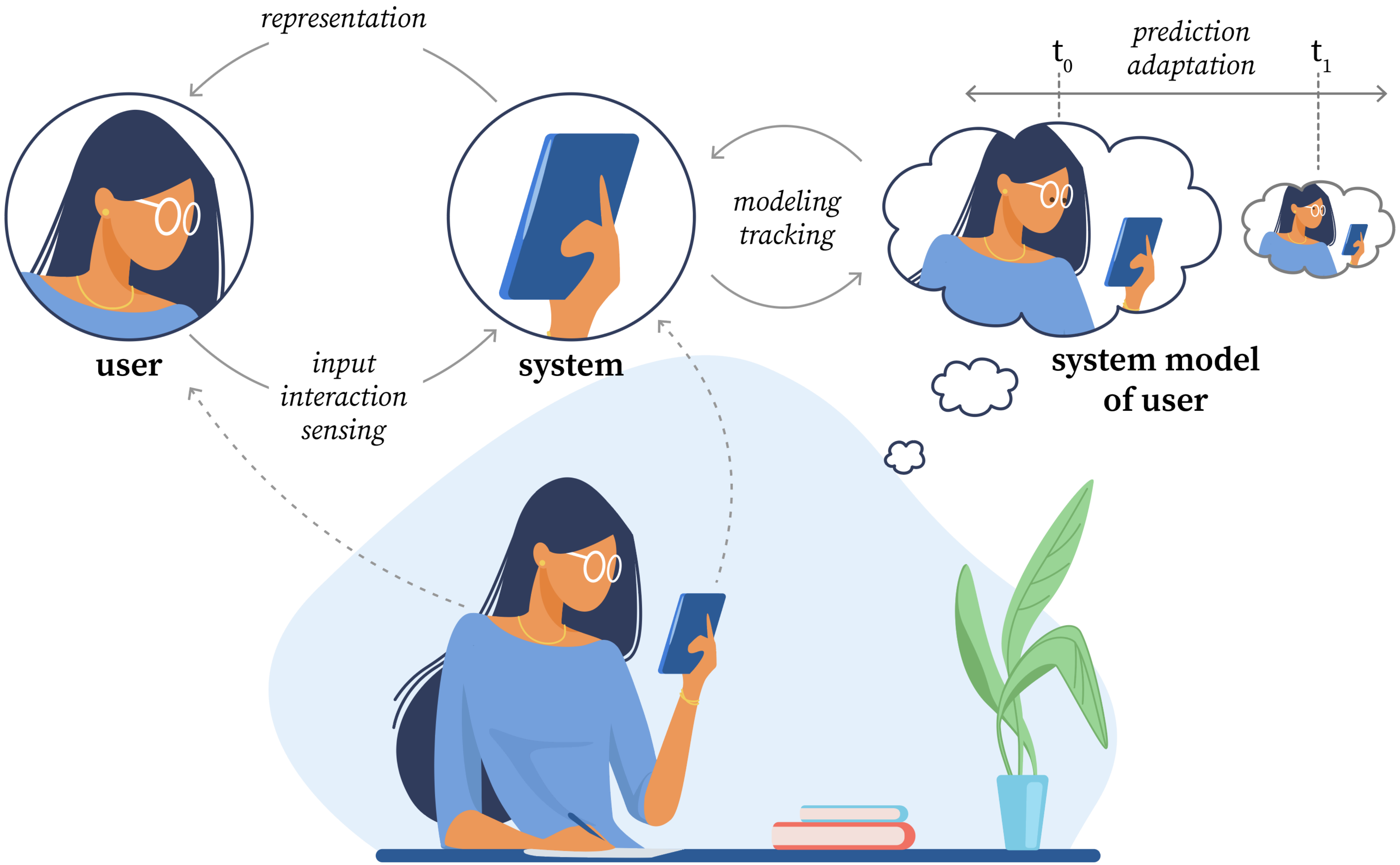


[5] # The Two UX Gulfs: Evaluation and Execution

# What do interactive systems look like?

*"a set of computer equipment and programs used together for a particular purpose… characterized by significant amounts of interaction [with] humans"*

→ A set of computer equipment and programs — components

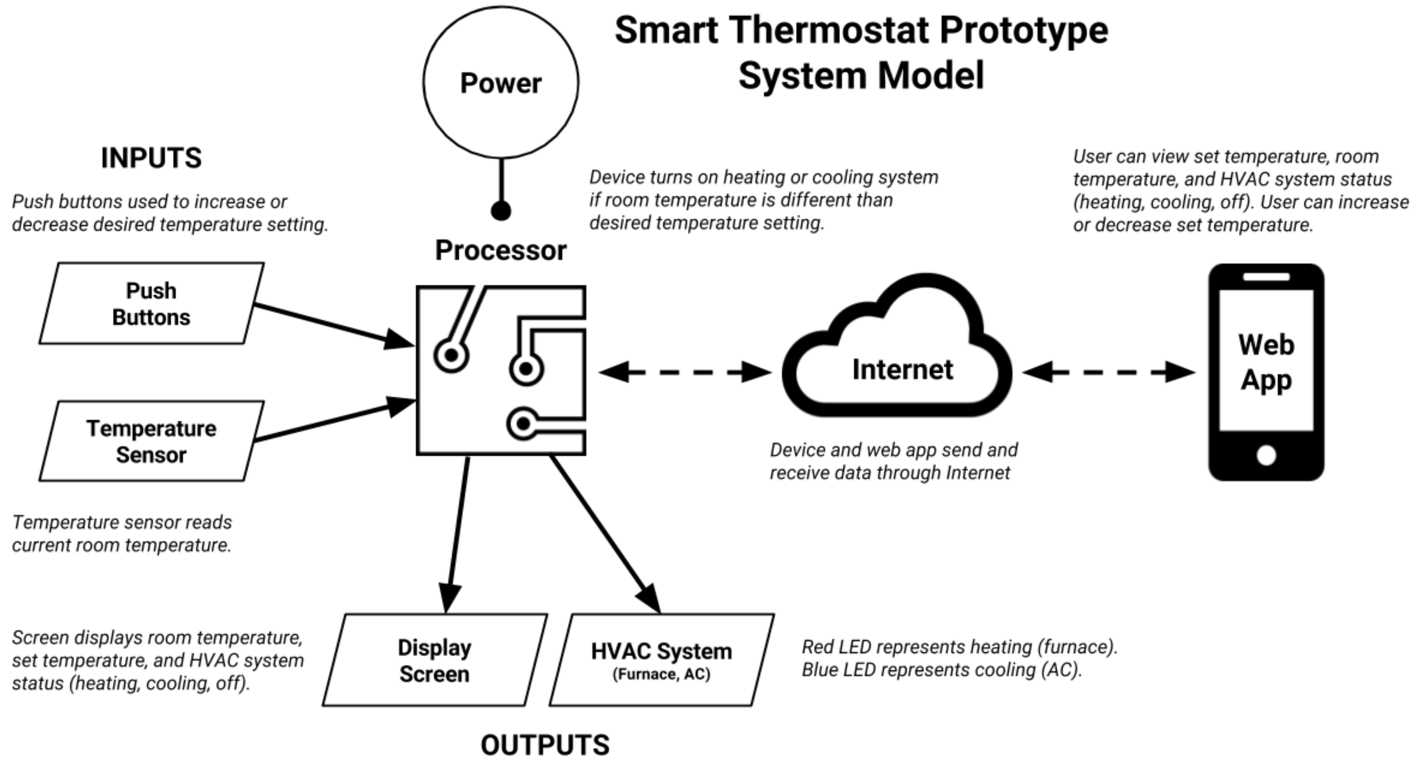


[6] Adapted from art by `pch.vector` / Freepik

# A Case Study

## Let's look at an example

# "Smart" Thermostat[7]



Smart Thermostat Prototype System Model

Power

Processor

INPUTS

Push buttons used to increase or decrease desired temperature setting.

Push Buttons

Temperature Sensor

Temperature sensor reads current room temperature.

Device turns on heating or cooling system if room temperature is different than desired temperature setting.

Internet

Device and web app send and receive data through Internet

User can view set temperature, room temperature, and HVAC system status (heating, cooling, off). User can increase or decrease set temperature.

Web App

Screen displays room temperature, set temperature, and HVAC system status (heating, cooling, off).

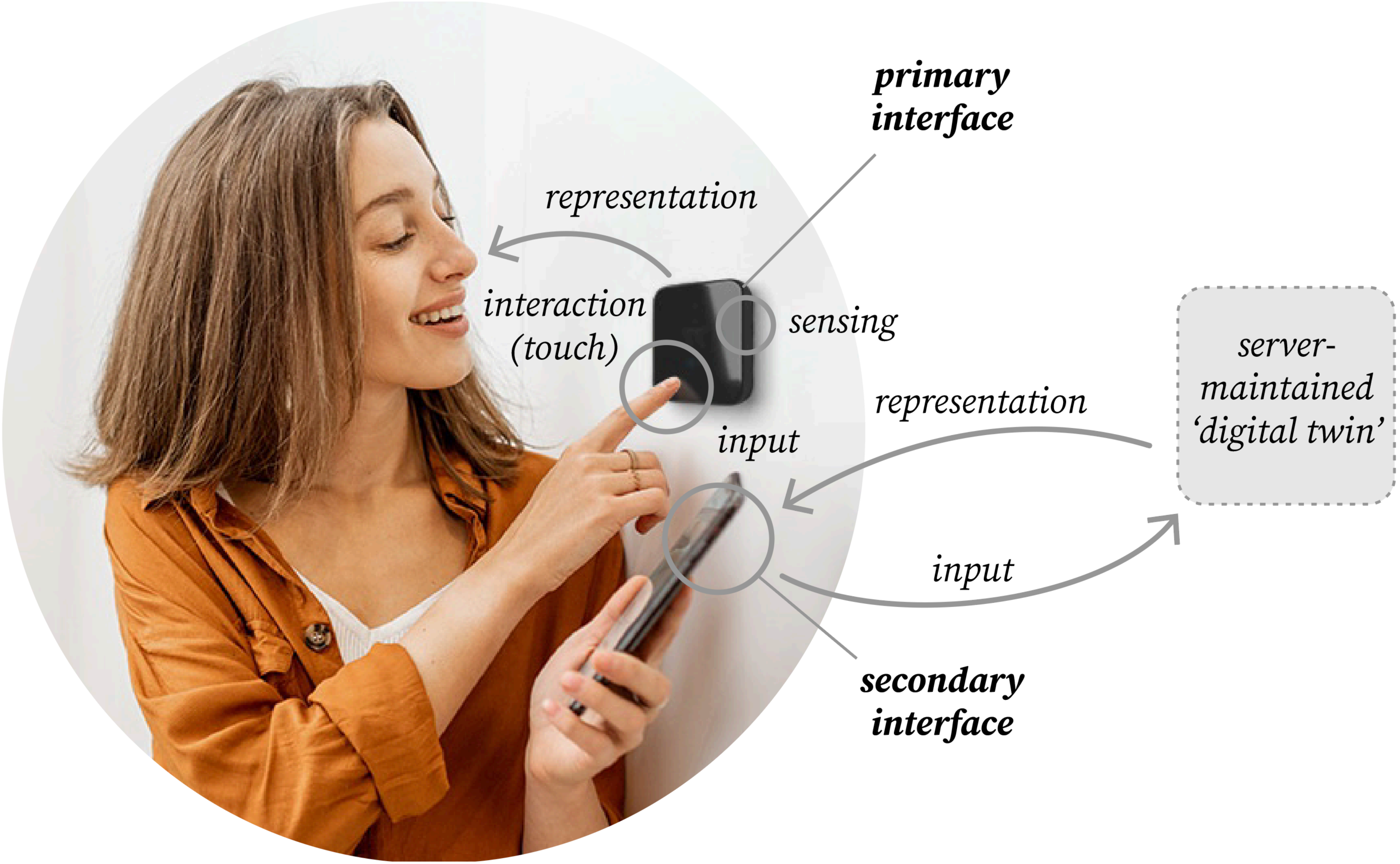
Display Screen

HVAC System (Furnace, AC)

Red LED represents heating (furnace). Blue LED represents cooling (AC).

OUTPUTS

[7] Images: Left, right



*representation*

***primary interface***

*interaction (touch)*

*sensing*

*input*

*representation*

*server-maintained 'digital twin'*

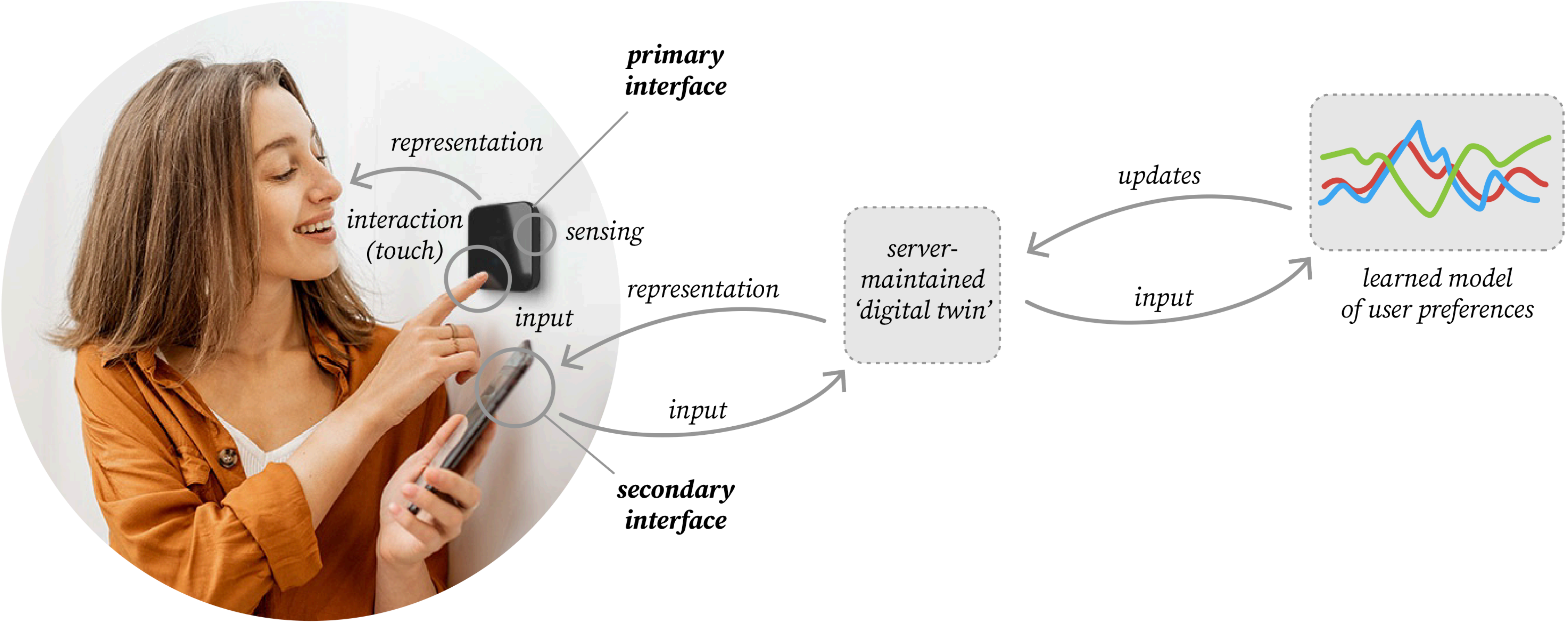
*input*

***secondary interface***

[8] Image source

Let's make it a bit more "smart."[9]

[9] Image source


primary
interface
representation
interaction
(touch)
sensing
input
representation
input
secondary
interface
server-
maintained
'digital twin'
updates
input
learned model
of user preferences

# An Exercise

# Identify A System

- → Form groups of 5, in 10 minutes:
  - → Pick an "interactive system" based on our definition (below)
  - → What type of system is it? Why?

An **interactive system** is...

*"a set of computer equipment and programs used together for a particular purpose... characterized by significant amounts of interaction [with] humans"*

# Questions?

# Course Information

From the course website:

*This course will introduce graduate students in computer science to core topics, methods, and principles related to building systems that interact with people. Students will learn from lectures; reading seminal and state-of-the-art papers on interactive systems; completing in-class and take-home assignments; and engaging a semester-long team project. The course will cover approaches, methods, devices, and algorithms for user sensing, decision making, and system behavior, integrating methods from machine learning, computer vision, robotics, and human-computer interaction.*

# Learning Objectives

1. Define interactive systems, identify system archetypes, and determine components
2. Gain familiarity with common approaches to sensing, decision making, representation, and user interaction
3. Gain familiarity with closed-loop, open-loop, and human-in-the-loop systems
4. Define new interactive systems based on application specifications
5. Prototype interactive systems and components using state-of-the-art tools, libraries, and frameworks
6. Design and conduct system and user evaluations
7. Write an “HCI systems” academic paper to present system design, technical specifications, and findings from evaluation

# Core Learning Objective — Building interactive systems

- → Ask yourselves, "what will be my **craft**?"
  - → **craft.** *skill in carrying out one's work*
- → How will you build what you will build?
  - → Focus on building **skill**
  - → **skill.** *the ability to do something well*
- → How will you build skill?
  - → Get comfortable with at least one platform/paradigm
  - → Practice, practice, practice

# Fit into the HCI Curriculum

Undergraduate HCI Courses:

1. CS-570 Introduction to Human-Computer Interaction — *focuses on methods*
2. CS-571 Building User Interfaces — *focuses on design & technical implementation*

Graduate HCI Courses:

1. CS/Psych/EdPsych-770 Human-Computer Interaction — *focuses on methods*
2. CS-839 Modeling Interaction — *focuses on people*
3. **CS-839 Building Interactive Systems** — *focuses on technology*

# Prerequisites

- → No formal prerequisites.
- → *Recommended:*[20]
  - → Fluency in at least one programming language (e.g., Python, C++)
  - → Familiarity with machine learning and/or computer vision libraries
  - → Experience working with sensors, mobile devices, and other embedded systems

[20] If you don't fit this profile, talk to me.

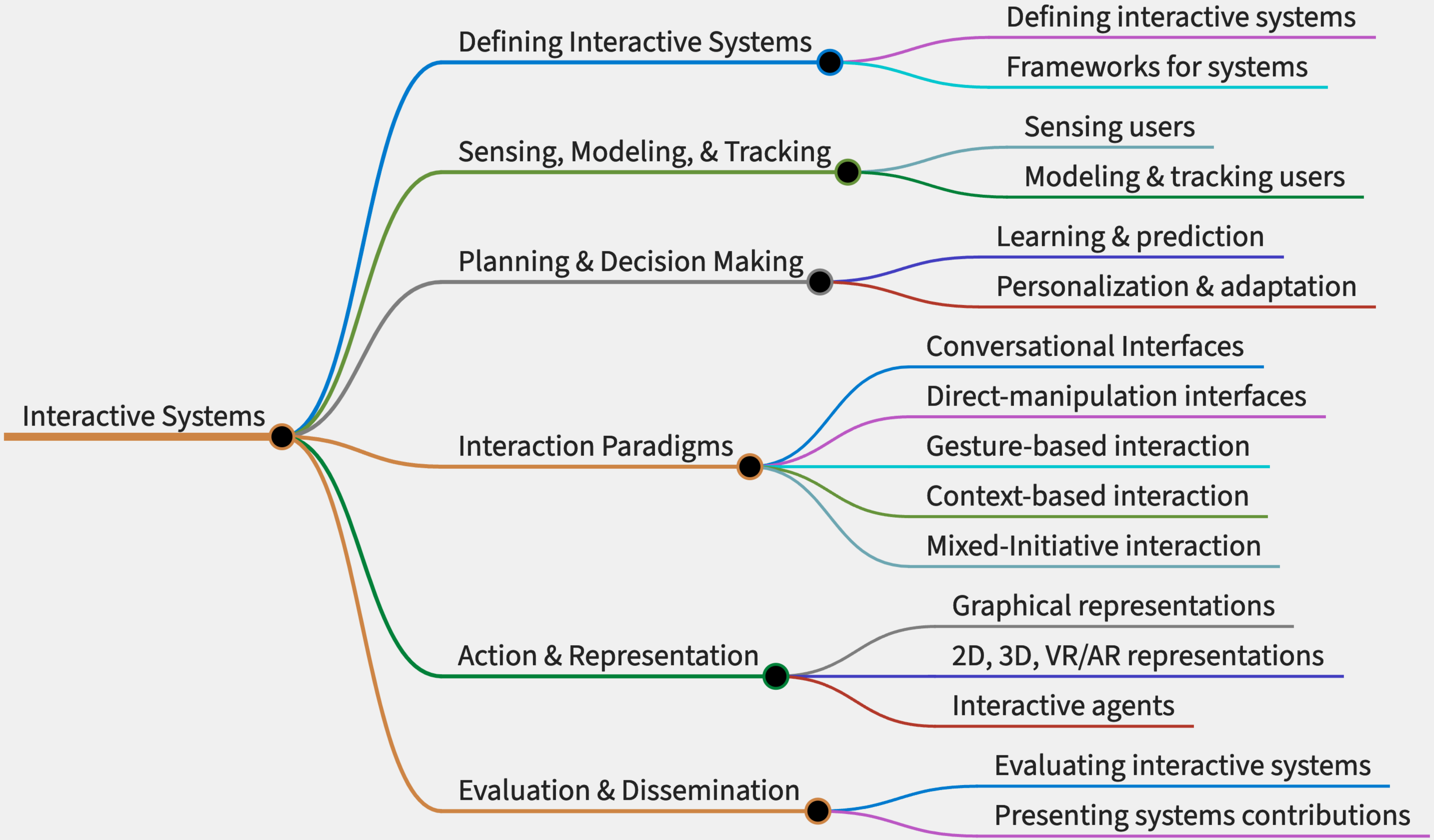
# Course Structure

Three modules:

1. **LEARN** — overview of each topic through lecture, readings, discussion
2. **HACK** — hands-on experience with topics through classroom activity, assignments
3. **INTEGRATE** — team projects that design, build, demo, evaluate an interactive system

# LEARN Module

- List of 15 topics (one per week), weekly reading (2–3 papers)
- **Reading groups**
  - 9 groups (5 members each) formed today (we can shuffle groups regularly)
  - Read and discuss before Monday class
  - Identify additional resources to bring into the discussion (a paper, a video, etc.)
  - Discussion can be in-person, on Piazza, or on Teams (reading groups choose)
  - Reading groups report back with three key points discussed (on Canvas, in class)



- Interactive Systems
  - Defining Interactive Systems
    - Defining interactive systems
    - Frameworks for systems
  - Sensing, Modeling, & Tracking
    - Sensing users
    - Modeling & tracking users
  - Planning & Decision Making
    - Learning & prediction
    - Personalization & adaptation
  - Interaction Paradigms
    - Conversational Interfaces
    - Direct-manipulation interfaces
    - Gesture-based interaction
    - Context-based interaction
    - Mixed-Initiative interaction
  - Action & Representation
    - Graphical representations
    - 2D, 3D, VR/AR representations
    - Interactive agents
  - Evaluation & Dissemination
    - Evaluating interactive systems
    - Presenting systems contributions

# HACK Module

- A total of 5 hack activities, each lasting 2–3 weeks
- **Hack teams**
  - 9 teams (5 members each) formed for each activity
  - Teams demo their hack on Wednesday classes

# INTEGRATE Module

- Semester-long substantial "system building" project
- **Project teams**
  - 9 teams (5 members each) formed on Week 3
  - Project milestones every 3-4 weeks
  - Teams demo milestones on Wednesdays
  - Final demo & paper submission at the end of semester

# Course Materials

→ Readings for next week will be available by Monday class

→ Lecture slides will be available immediately before class

→ Everything except work you do will be open-source, publicly available

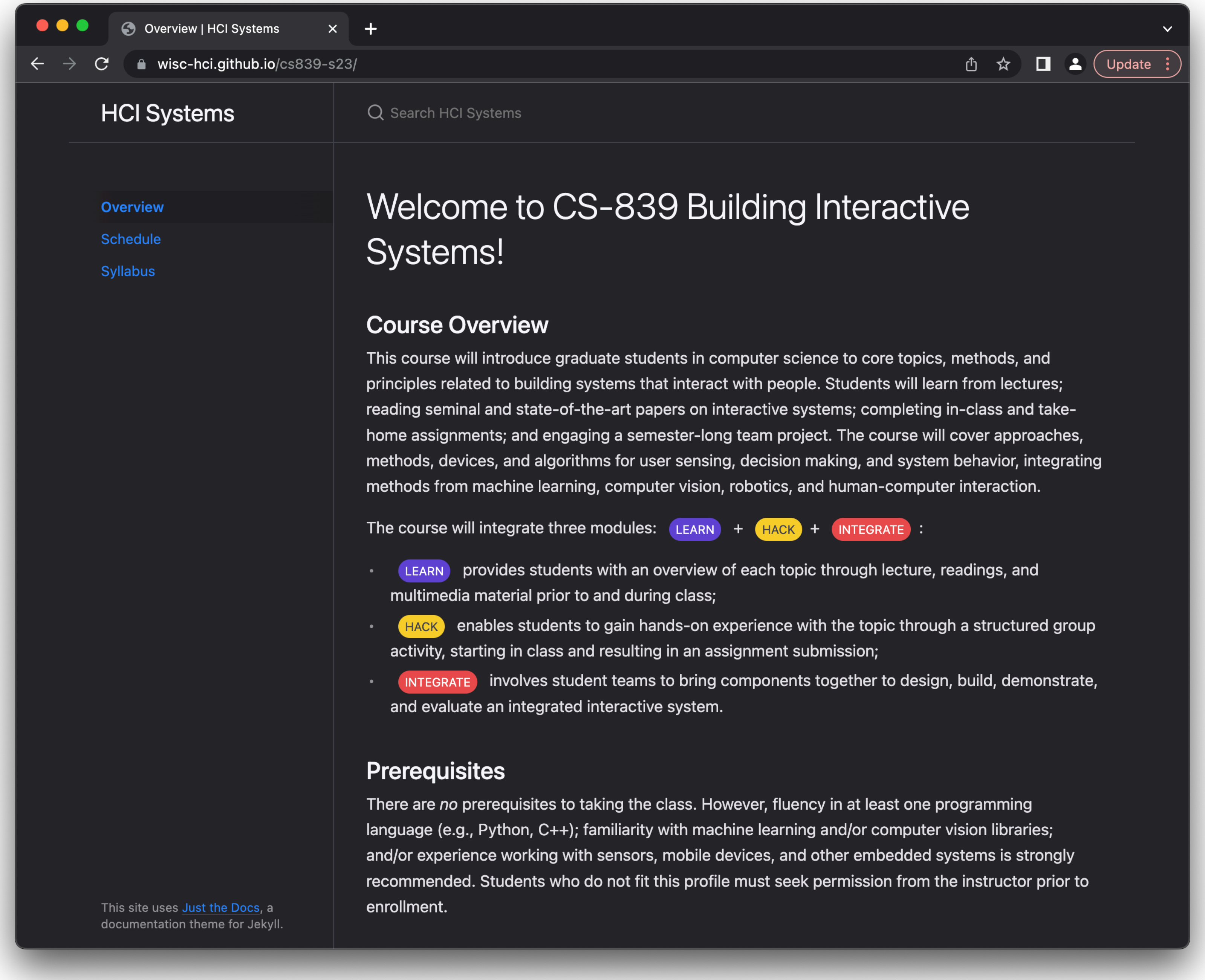
# Grading

**Rule of thumb:** Don't worry about your grade. If you do everything you are asked to do, you will get an A or AB. Focus on exploring, learning, and having fun.

**Ancillary:** What you learn at grad school will be the foundation of the skills and knowledge you will use for the rest of your careers, and you will never have the kind of time and ability to focus again, so take advantage of it.

# Course Resources

- → Where to start:
  - → Course website: https://wisc-hci.github.io/cs839-s23/
- → For assignments, submissions:
  - → Canvas: https://canvas.wisc.edu/courses/345265
- → For questions:
  - → Piazza: https://piazza.com/wisc/spring2023/cs839
- → Office hours:
  - → Right after each class: MW (2:15-3:00pm)



Welcome to CS-839 Building Interactive Systems!

Course Overview

This course will introduce graduate students in computer science to core topics, methods, and principles related to building systems that interact with people. Students will learn from lectures; reading seminal and state-of-the-art papers on interactive systems; completing in-class and take-home assignments; and engaging a semester-long team project. The course will cover approaches, methods, devices, and algorithms for user sensing, decision making, and system behavior, integrating methods from machine learning, computer vision, robotics, and human-computer interaction.

The course will integrate three modules: LEARN + HACK + INTEGRATE :

- LEARN provides students with an overview of each topic through lecture, readings, and multimedia material prior to and during class;
- HACK enables students to gain hands-on experience with the topic through a structured group activity, starting in class and resulting in an assignment submission;
- INTEGRATE involves student teams to bring components together to design, build, demonstrate, and evaluate an integrated interactive system.

Prerequisites

There are *no* prerequisites to taking the class. However, fluency in at least one programming language (e.g., Python, C++); familiarity with machine learning and/or computer vision libraries; and/or experience working with sensors, mobile devices, and other embedded systems is strongly recommended. Students who do not fit this profile must seek permission from the instructor prior to enrollment.

This site uses Just the Docs, a documentation theme for Jekyll.

# Course Feedback

- This is a *brand new* course
- I will need your feedback
  - Improve the course as we go
  - Improve it for future offerings
- Feedback formats
  - Persistent Piazza thread for anonymous feedback
  - Regular reminders to seek feedback, in-class discussion

# Questions?

# Next Week

- Due Monday
  - Readings, reading group commentary submission on Canvas
- Monday class
  - Lecture on *interactive systems frameworks*
- Wednesday class
  - In-class activity on *modeling systems*