

# CS 349 Winter 2025

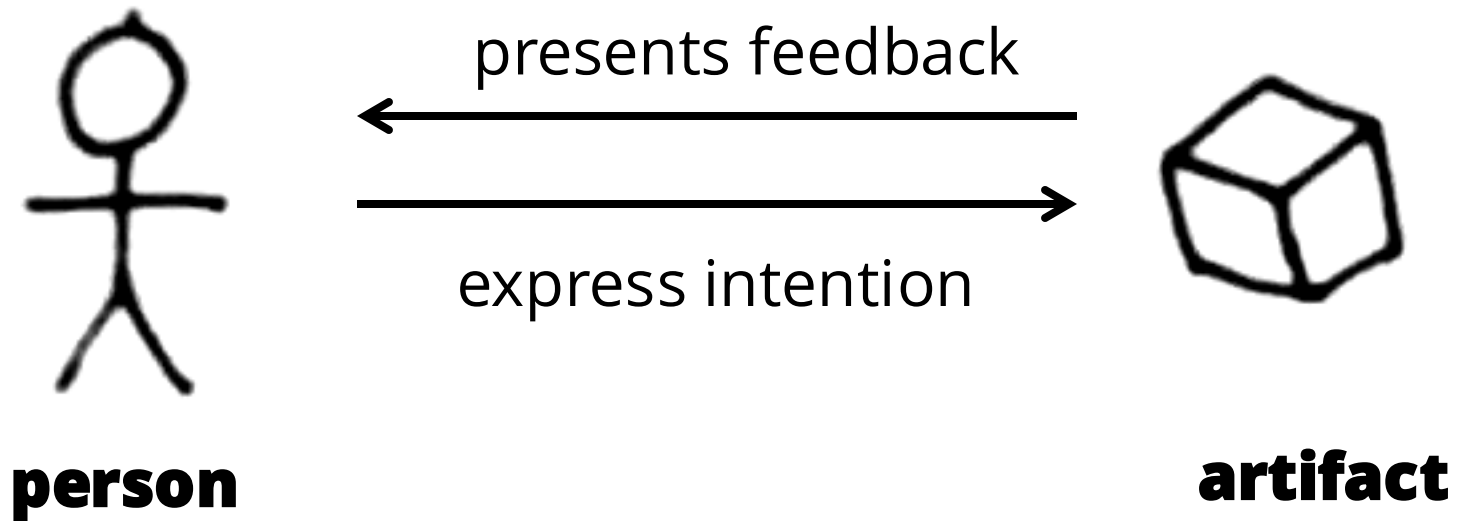
Instructors: Matthew Brehmer (001) and Adrian Reetz (002, 003)

<https://student.cs.uwaterloo.ca/~cs349/1251/>

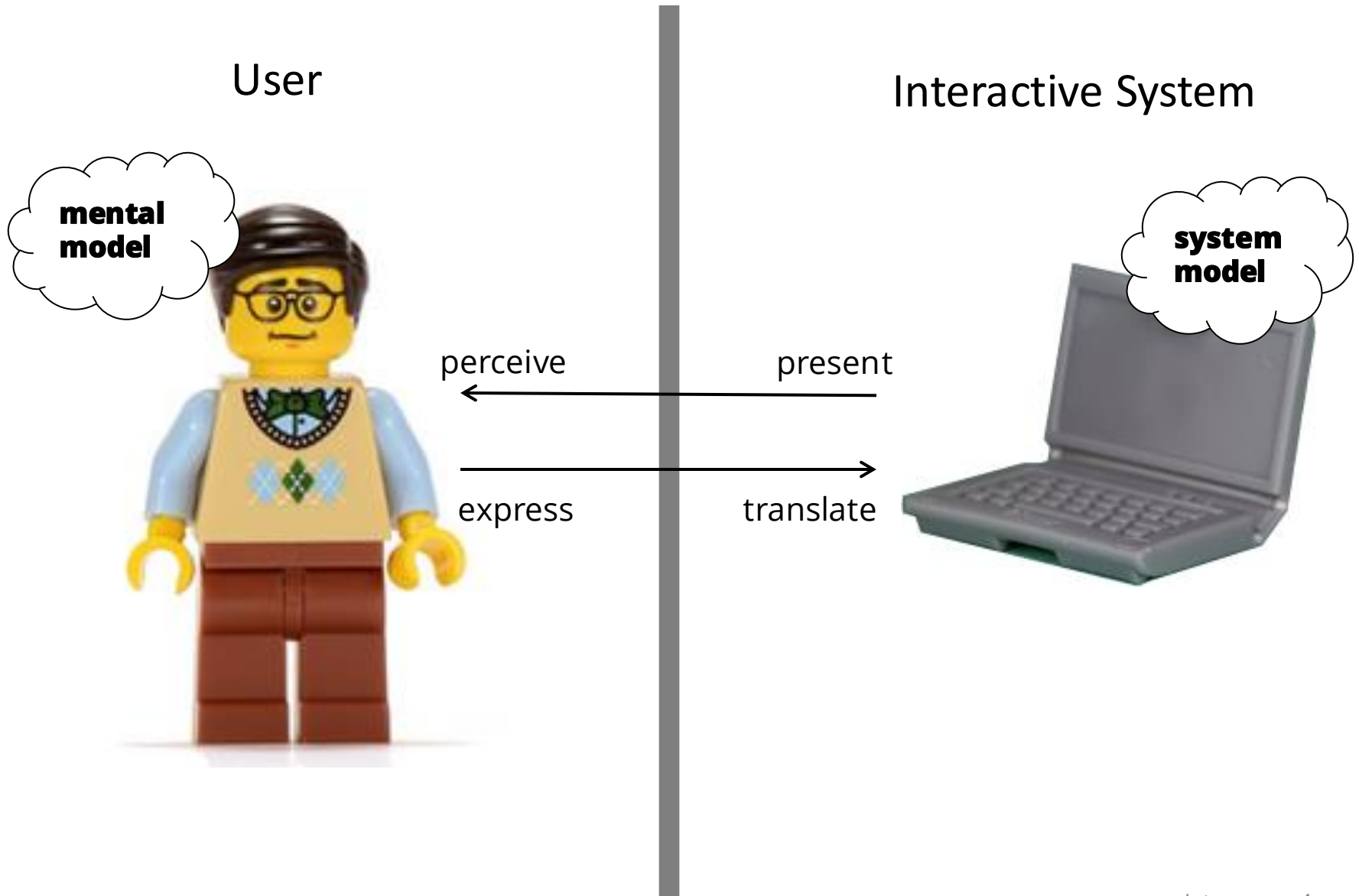
computer

# User Interface

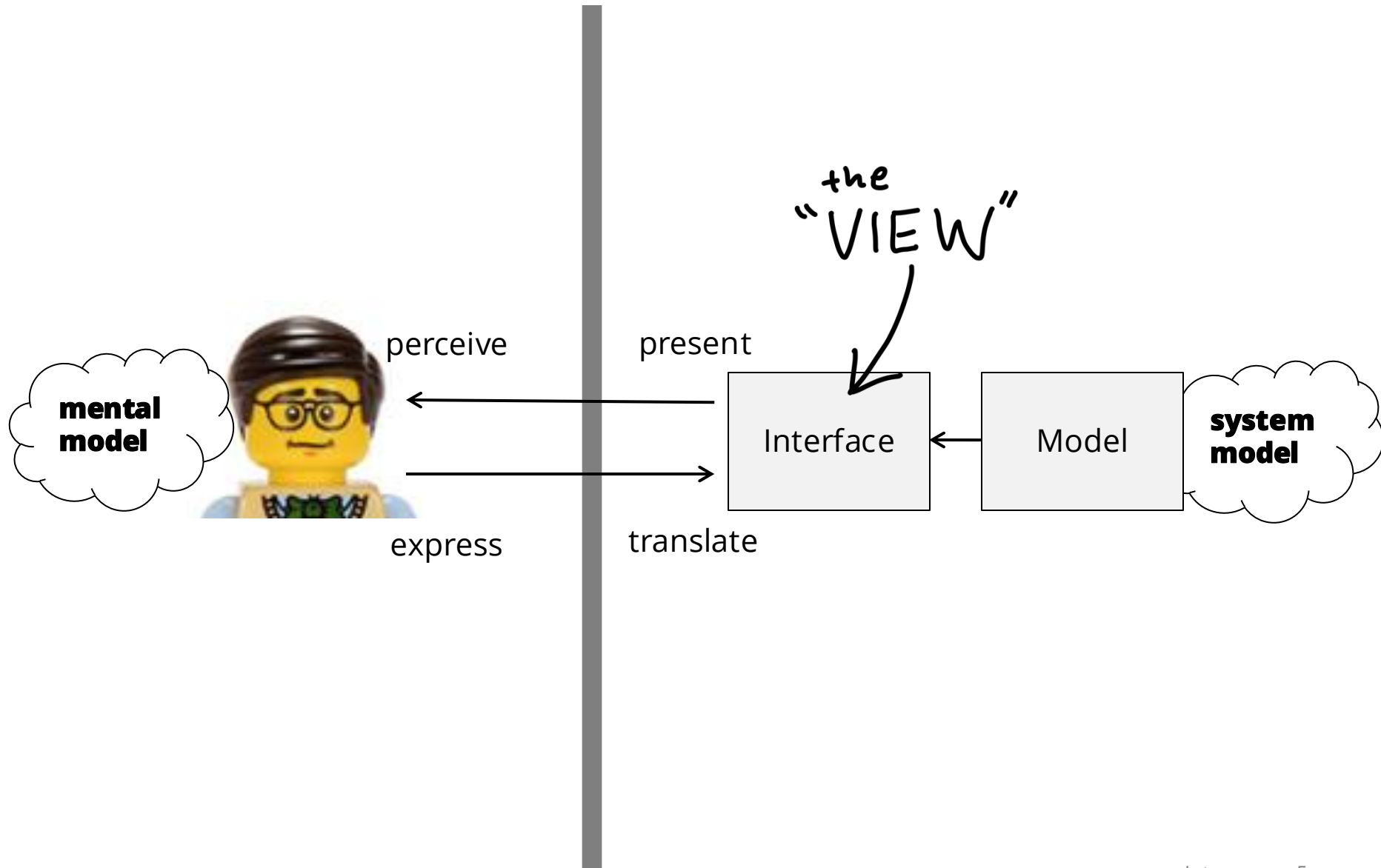
- The human's view of a computer
- *Formally*: The place where a person expresses intention to an artifact, and the artifact presents feedback to the person.



# Interactive System Architecture

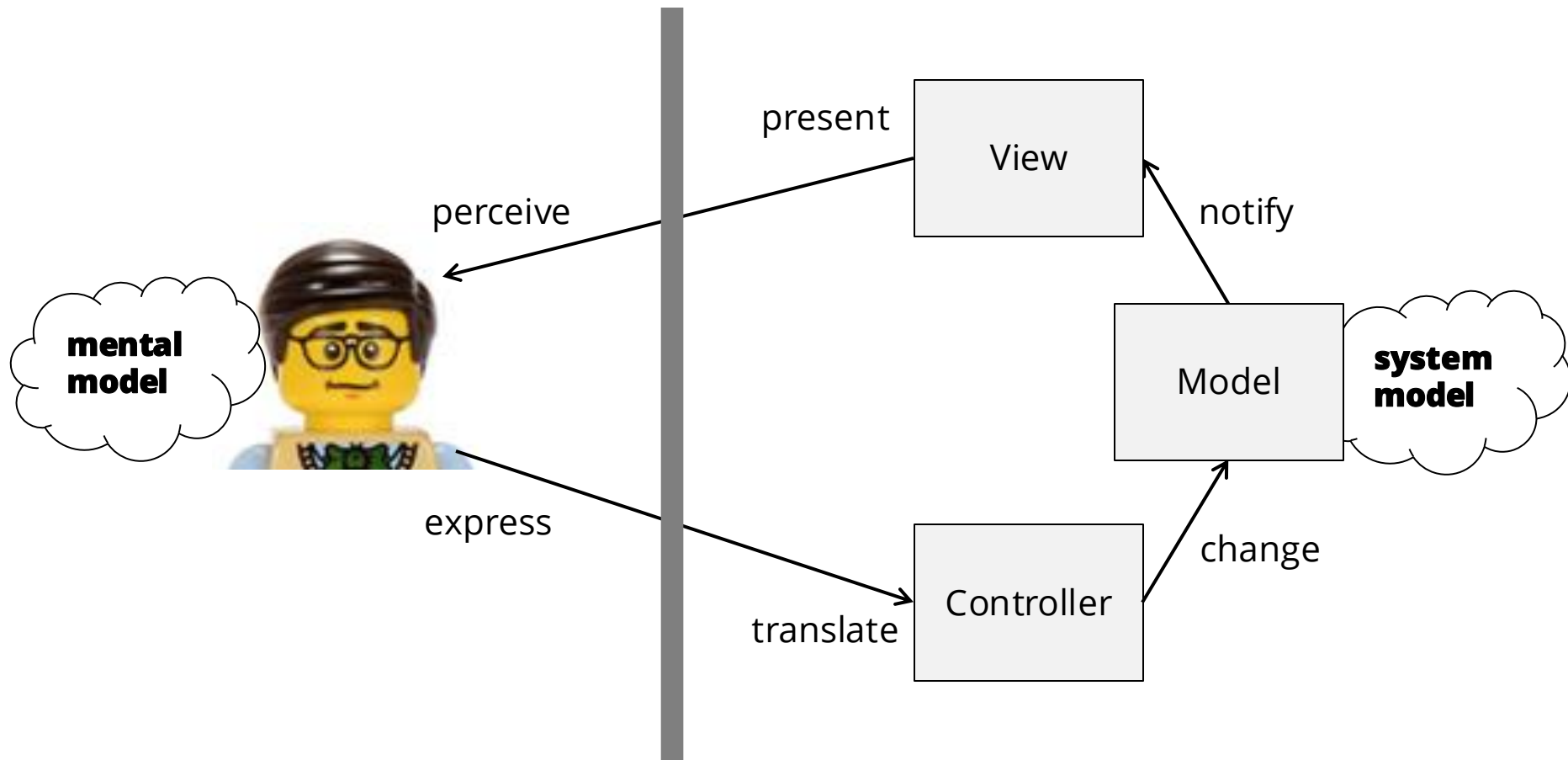


# Interactive System Architecture

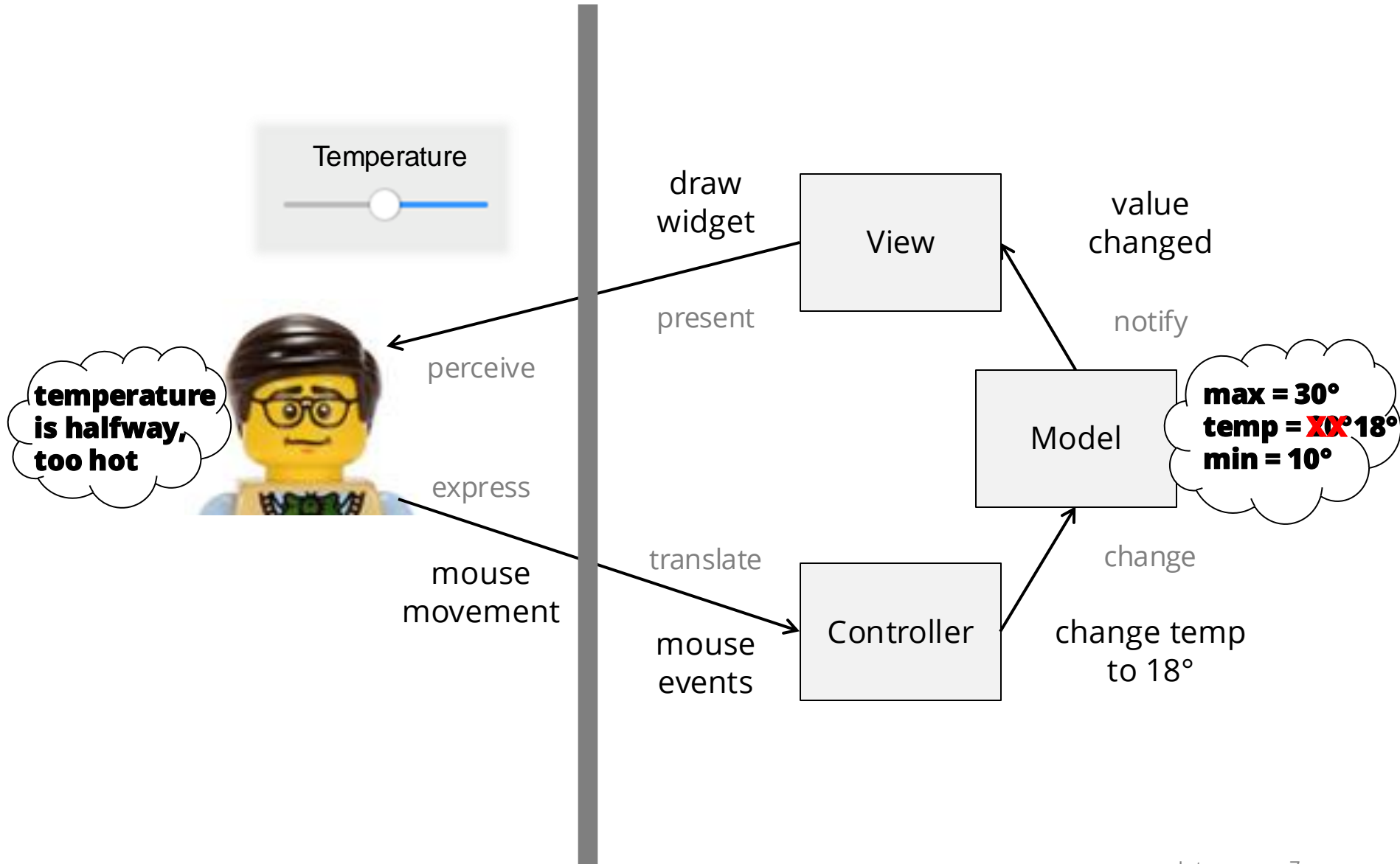


# Model-View-Controller (MVC)

MVC was the first MV\* interactive system architecture

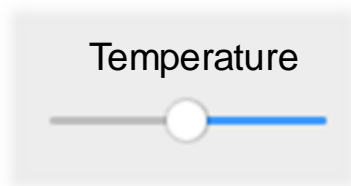


# Graphical Temperature Control



# User Interface vs. User Interaction

- **Interface** refers to the external presentation to the user
  - Controls (what user manipulates to communicate intent)
  - Feedback (what the program uses to communicate its response)
- **Interaction** refers to actions by user and system over time
  - interaction is a dialog with a cycle alternating between the user manipulating controls and the system responding with feedback



widget = control + feedback

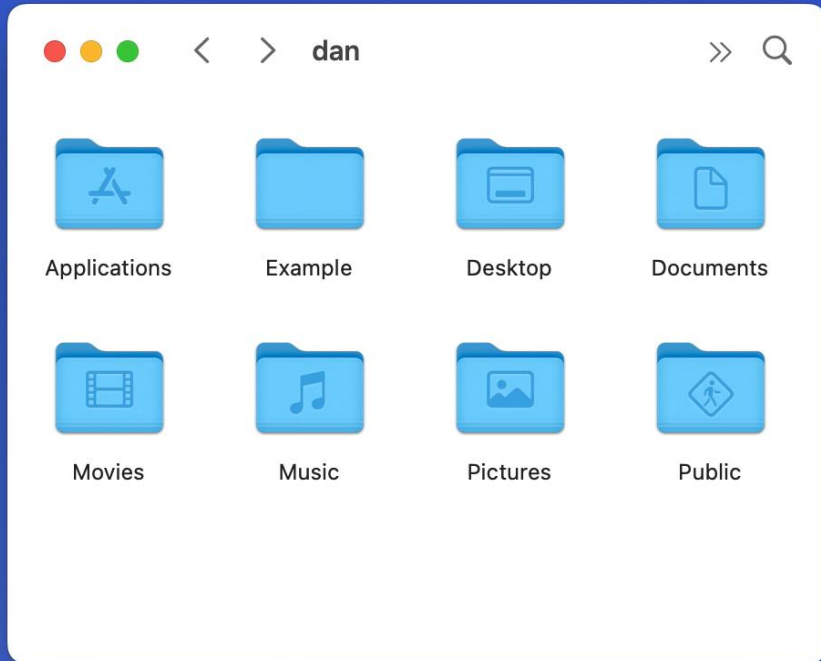


# Graphical User Interface (GUI)

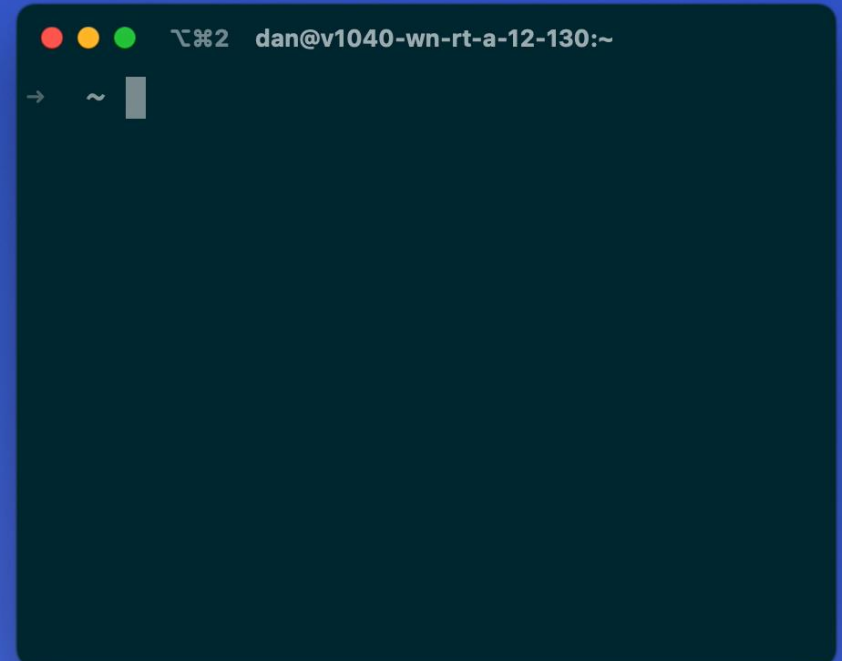
- We can assume:
  - a **pointing device** (e.g. mouse)
  - a **text entry device** (e.g. keyboard)
  - a **high-resolution display**
- The display contains interactive elements (e.g. widgets)
- Users interact primarily by pointing and clicking
  - pointing at an object of interest (e.g. widget, image, text)
  - clicking to select, drag to move, etc.



# GUI actions have equivalent system commands



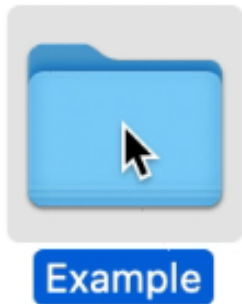
**GUI**



**Command Line**

# Advantages of a GUI interface

1. Emphasizes **recognition** over **recall** of available actions
  - Easier to discover options and experiment
2. Uses **metaphor** and **analogy** to make interface more intuitive
  - Usage is closer to how things are done in real works  
e.g. “desktop”, “folder”, “drag”, “drag-and-drop”, “icons”, ...

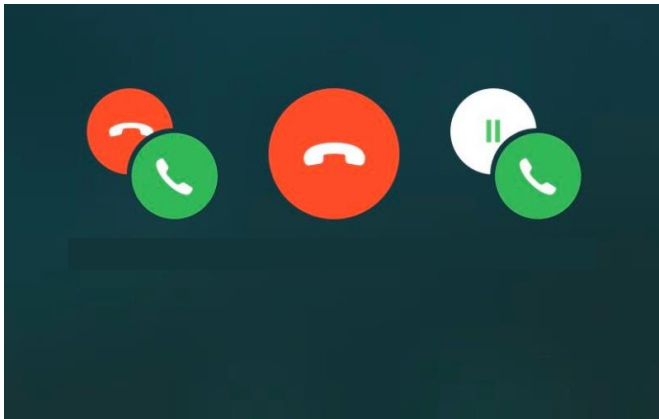


vs.

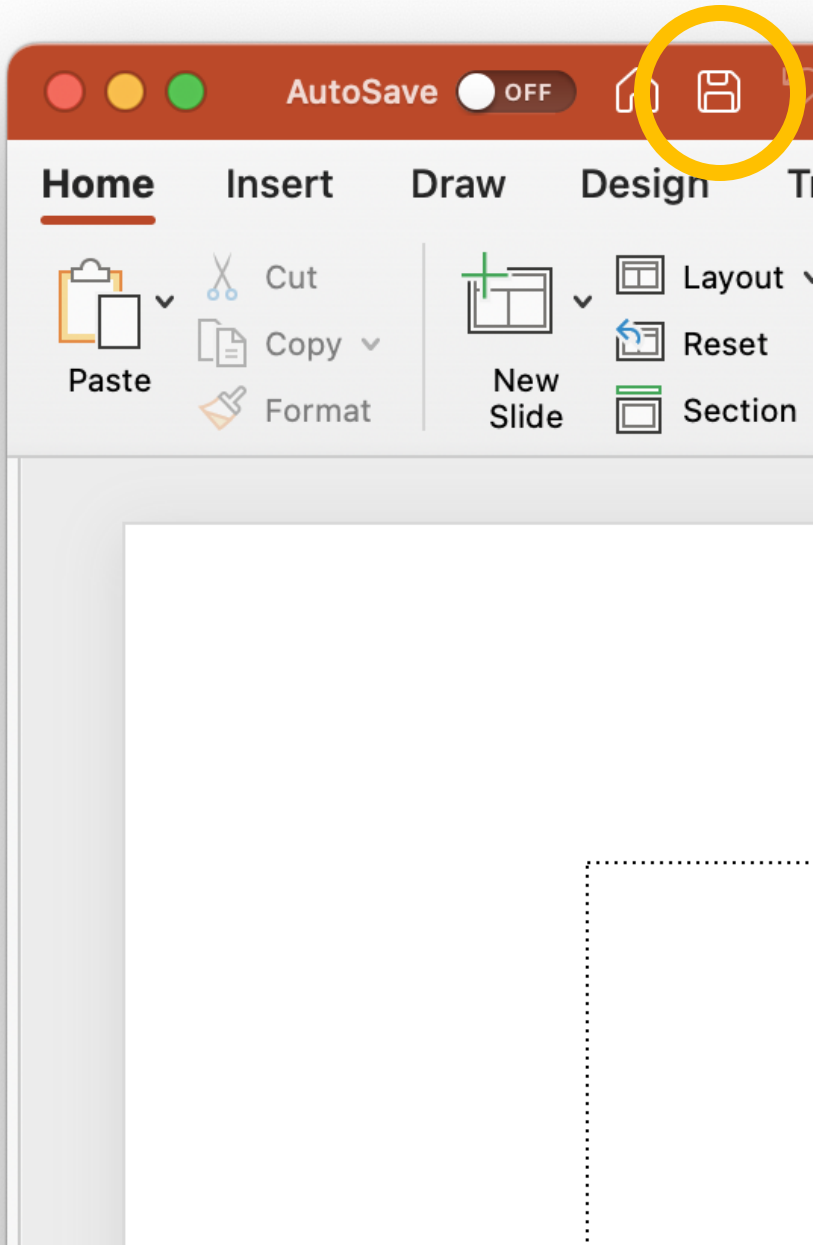
```
→ ~ cd Example/
```

# Disadvantages of Graphical User Interfaces

- Consumes valuable screen space, forcing information off-screen
- Switching between keyboard and pointing device time consuming
- Visual representations may not be clear



- Visually impaired users can't see the graphics; no linear flow for screen readers; physically impaired may have difficulty with required movements

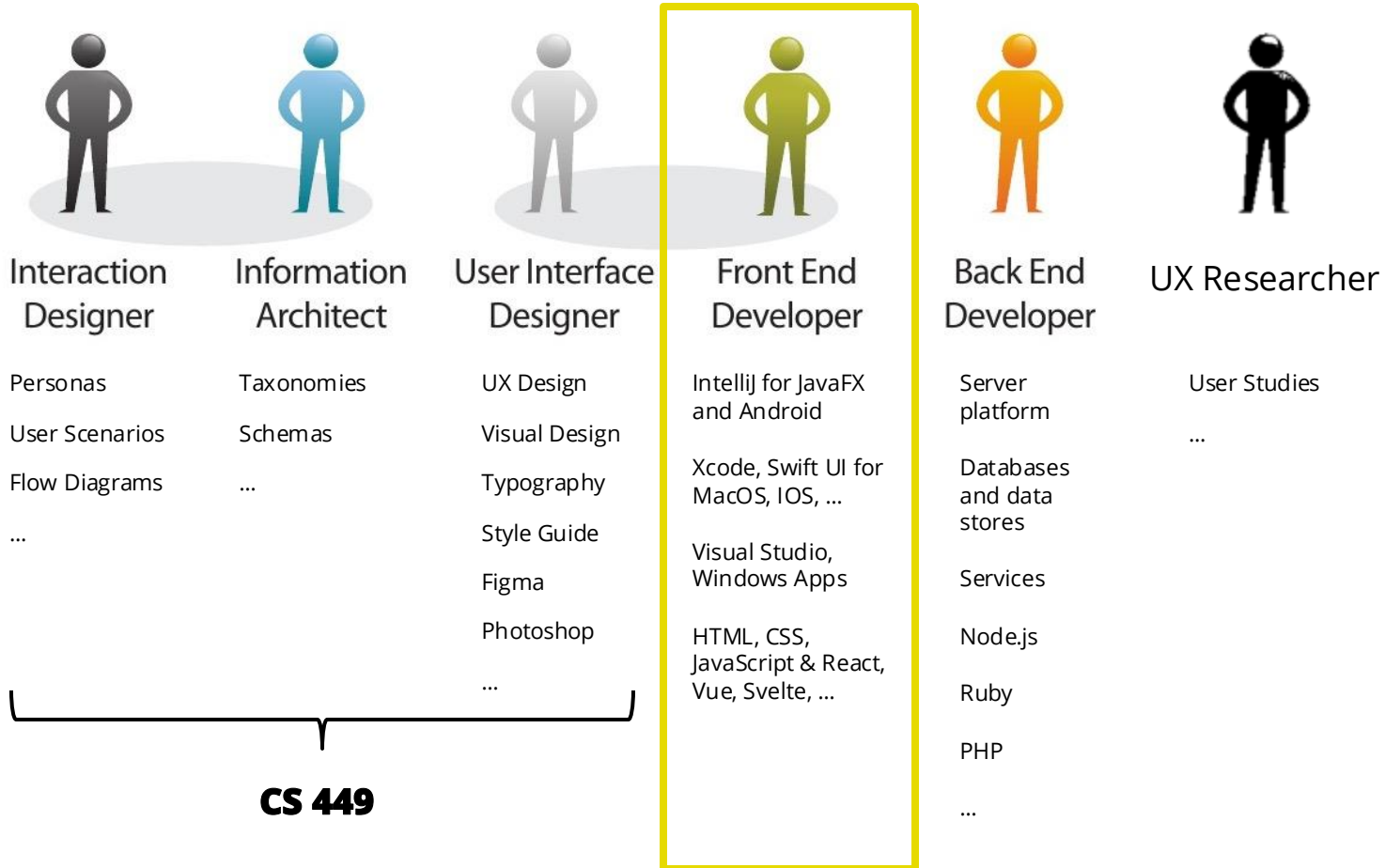


# Course Information

- Focus
- Technology
- Syllabus

# UI Development in Industry

## CS 349



# What will you learn?

The focus is *how user interfaces work* and *how to build user interfaces*, using **web technologies and the TypeScript language** 🍕

## Learning Outcomes

- Explain architectural and algorithmic details underlying user interfaces and user interface toolkits
- Describe key aspects of user interfaces, such as user input, event-driven architecture, events, etc.
- Implement user interfaces using different approaches

This course is not a tutorial for a programming language or UI toolkit

This course is technical with an emphasis on programming



# Course Website

- <https://student.cs.uwaterloo.ca/~cs349/1251/>

The screenshot shows a dark-themed web browser window. The browser tab is titled "CS349" and the address bar shows "https://student.cs.uwaterloo.ca/~cs349/1251/". The website header includes a logo with a large "U" and "CS349" text, and the course title "CS349-W25". A search bar is visible with the placeholder text "Search...". The left sidebar contains a navigation menu with links for "Home", "Assignments", "Policies", "Schedule", and "Support". At the bottom of the sidebar, there is a "Auto" button and a footer that says "Built with ❤️ by Hugo". The main content area features the course title "CS349" in large white text. Below the title is a red warning banner with a white triangle icon and the text "Warning". Underneath the banner, a bolded message states: "This website is a draft: all information is tentative and subject to change." The main content area also includes an introductory paragraph: "An introduction to contemporary user interface implementation concepts, including event abstraction, graphical components, layout, feedback, testing, accessibility, and architectures to develop user interfaces. One or more types of interface toolkit paradigms are considered. (For a full description, please see the course calendar.)" Below this is a section titled "Learning Outcomes" with a bulleted list of three items: "Explain architectural and algorithmic details underlying current user interfaces and user interface toolkits.", "Describe key aspects of user interfaces, such as user input, event-driven architecture, asynchronous events, etc.", and "Implement user interfaces in a browser with Typescript: (1) using the HTML Canvas; (2) using HTML elements and DOM manipulation, and (3) using a modern UI web development".

# Website Highlights

- <https://student.cs.uwaterloo.ca/~cs349/1251/>
- About
  - Course Staff: ISC, IA, TAs
  - Course Communication: Piazza (hours), Office Hours
  - Assessment: 5 assignments, midterm, final
- Schedule
- Assignments
  - **Develop on your computer**, submit using git
  - Late submissions
- Policies
  - **Contact Caroline Kierstead (ISC) if sick**
  - Short-term absence: **must also complete course form**
  - Academic integrity and external sources (incl. AI systems)

A0 will setup your dev environment



# Midterm and Final Exam

- Structure
  - terms and definitions
  - UI toolkit algorithms, architectures, and usage
  - reading and writing short code (e.g. what is the result of this code, fill in the blank, write the function body, write short program, etc.)
  - multiple choice, short answer, longer questions
- How to do well:
  - attend lectures
  - pay attention and take notes
  - do the assignments