CS 349: User Interfaces

https://www.student.cs.uwaterloo.ca/~cs349

Jeff Avery & Alex Williams

Winter 2017

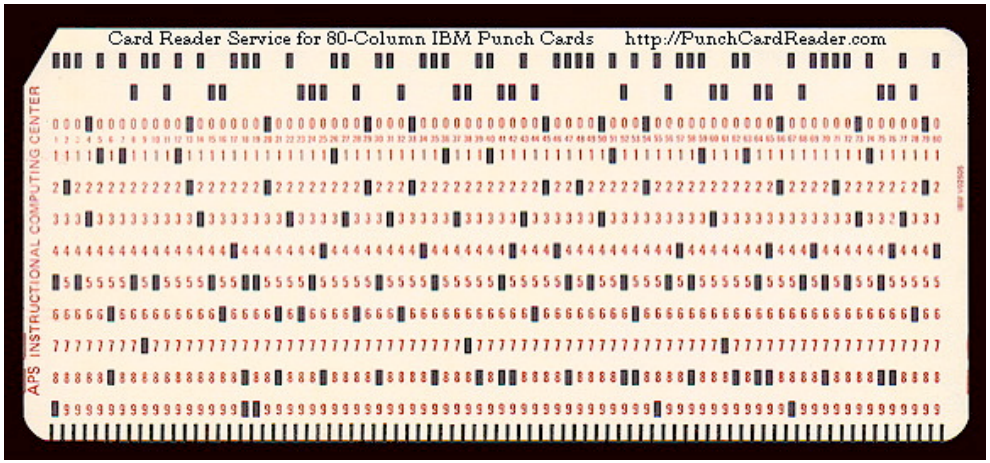
Pop Quiz

- Why is Waterloo so well-known in Canada for Computer Science?
- Answer: Because 40 years ago, this is what a computer looked like …

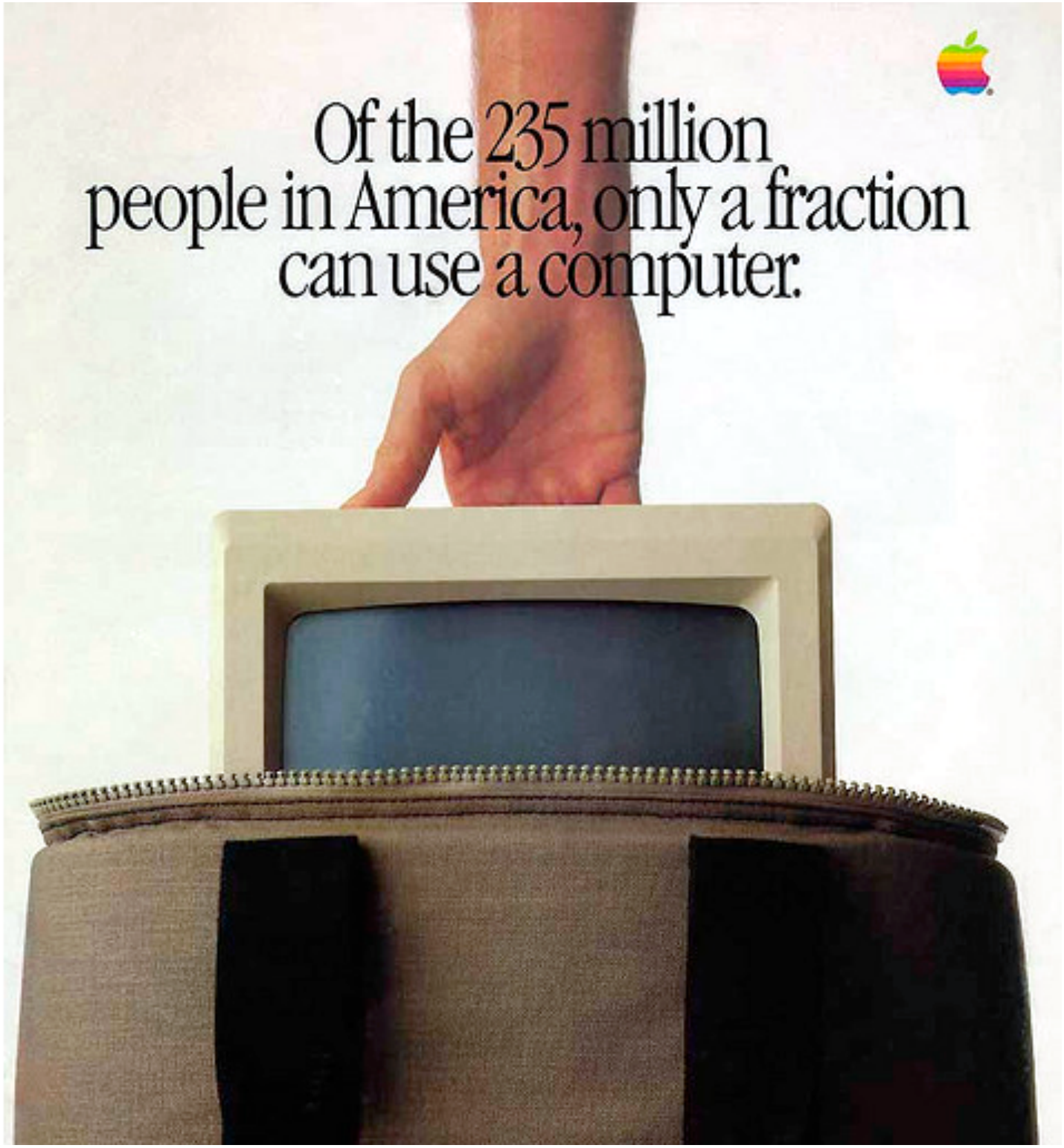
Red Room, October 1974

Another Way to Think About This

- This is what a user-interface looked like before the early 1980s:

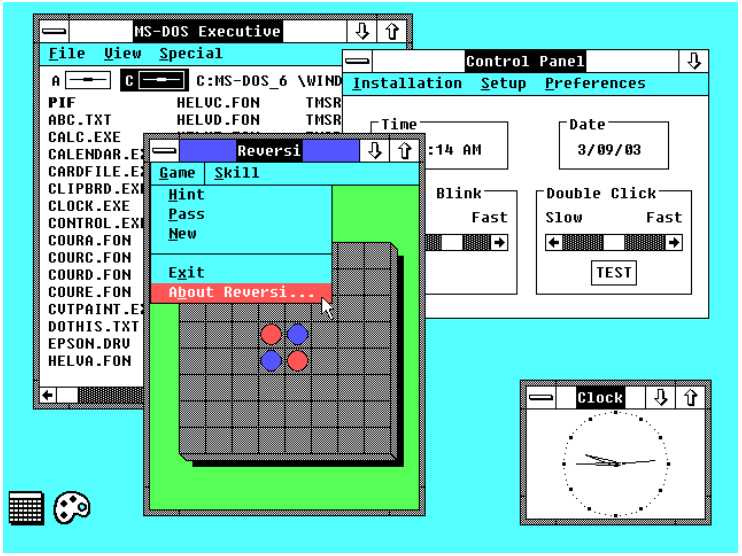


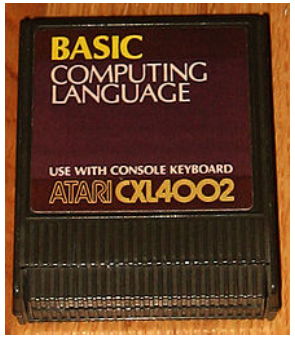

```
                        SAMPLE APPLICATION FORM

APPLICATION NO : _______

                ------------------------------------------
               |READ DETAILED INSTRUCTIONS GIVEN SEPARATELY|
               |    BEFORE FILLING THE APPLICATION FORM.   |
                ------------------------------------------

  NAME OF THE APPLICANT :  _________ __________ __________
                           FIRSTNAME  MIDDLE     LAST-NAME
  DATE OF BIRTH : __ / __ / ____

  RESIDENTIAL ADDRESS :  ____________________________
                         ____________________________

  EDUCATIONAL DETAILS
  ------------------------------------------
  |QUALIFICATION | UNIVERSITY     | YEAR   |
  ------------------------------------------
  | _________    | _________      | ____   |
  | _________    | _________      | ____   |
  | _________    | _________      | ____   |
  ------------------------------------------
```

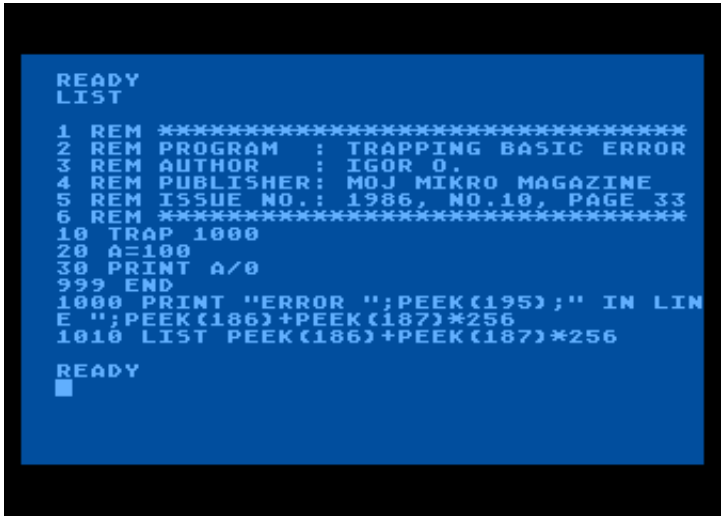

Of the 235 million
people in America, only a fraction
can use a computer.


hello.

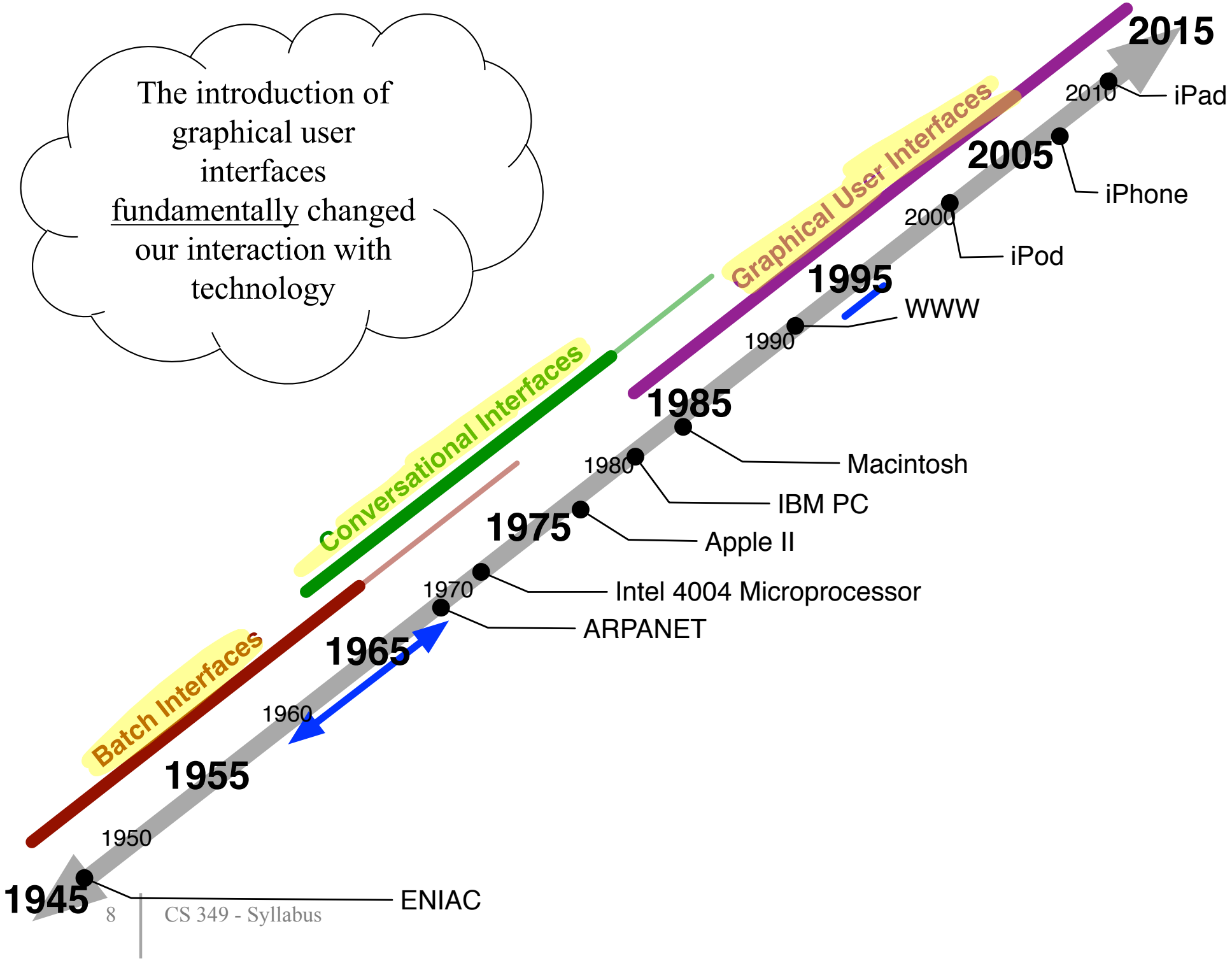
Computing in the 1980s







Computing Today


The introduction of graphical user interfaces fundamentally changed our interaction with technology
Batch Interfaces
Conversational Interfaces
Graphical User Interfaces
1945
ENIAC
1950
1955
1960
1965
1970
ARPANET
Intel 4004 Microprocessor
1975
Apple II
1980
IBM PC
1985
Macintosh
1990
1995
WWW
2000
iPod
2005
iPhone
2010
iPad
2015

How Has Computing Changed?

	40 years ago	Today
Who used a computer	Specialist	Everyone
What they knew about the computer	Lots	Little
How they learned to use a computer	Reading operators manuals	Tinkering
Where they found the applications that ran on the computer	Built them	App store, web
How many different computers they would have interacted with during the course of their day	*Maybe* one	Many!
Primary mode of interaction	Punch-cards, command-line, menus	Graphical, touch, speech

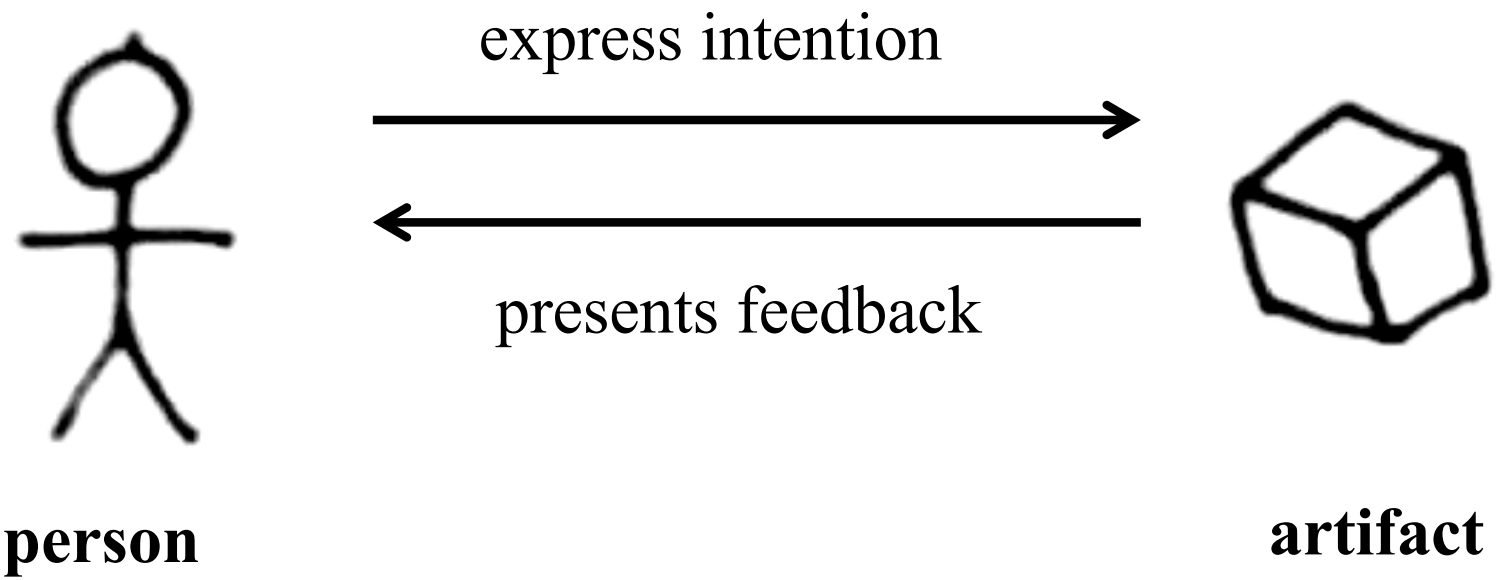
What is a User Interface?

- Maybe, “the human’s view of the computer”

- Maybe, “the place where humans and computers meet”

Definition: User Interface

- A user interface is the place where a person *expresses intention* to an artifact, and the artifact *presents feedback* to the person.
- Essentially, it's the way that *people* and *technology* interact.



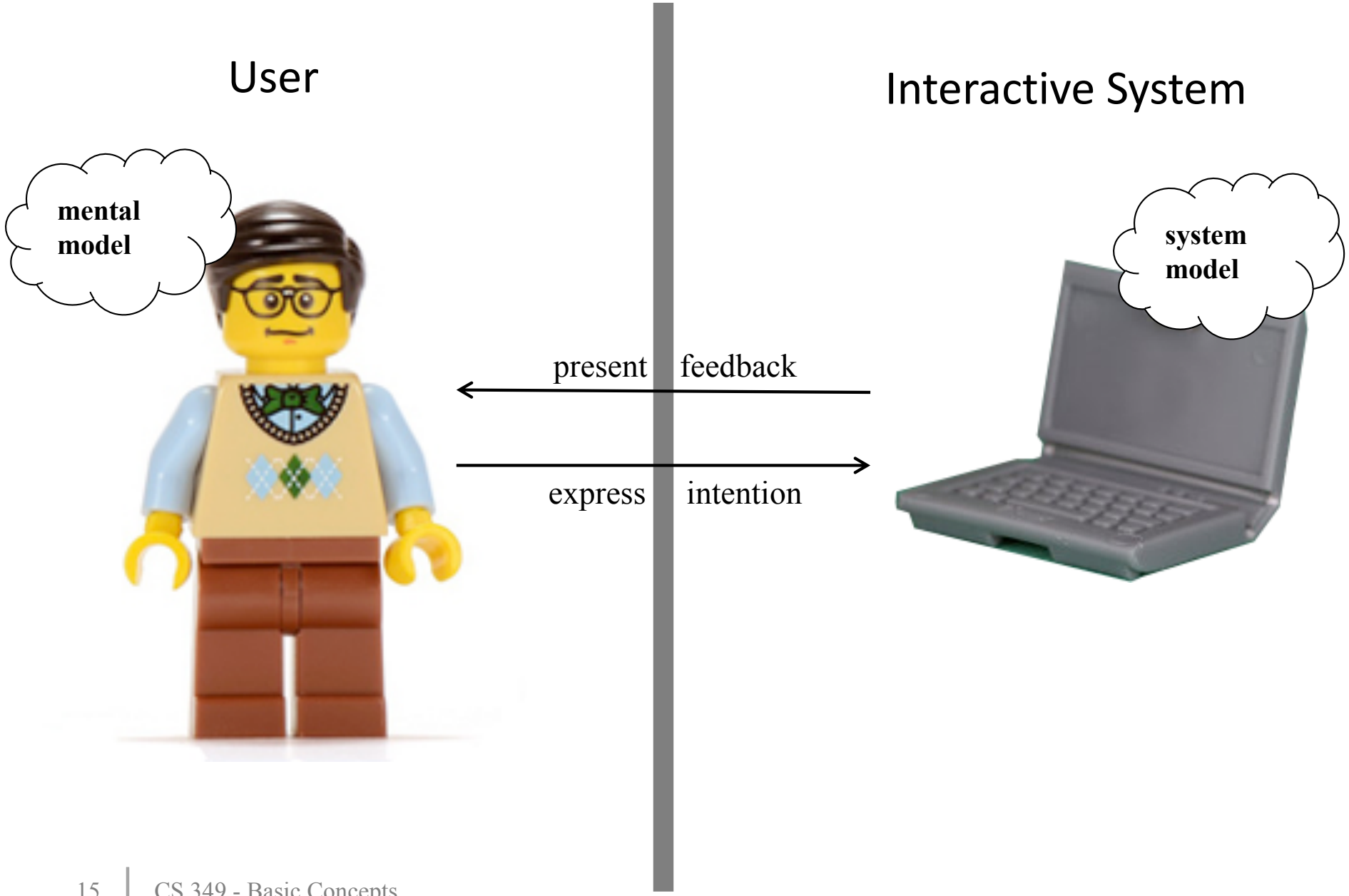
User Interfaces

- Does a microwave have an interface?
- A refrigerator?
- A door bell?
- A hammer?
- A jet?

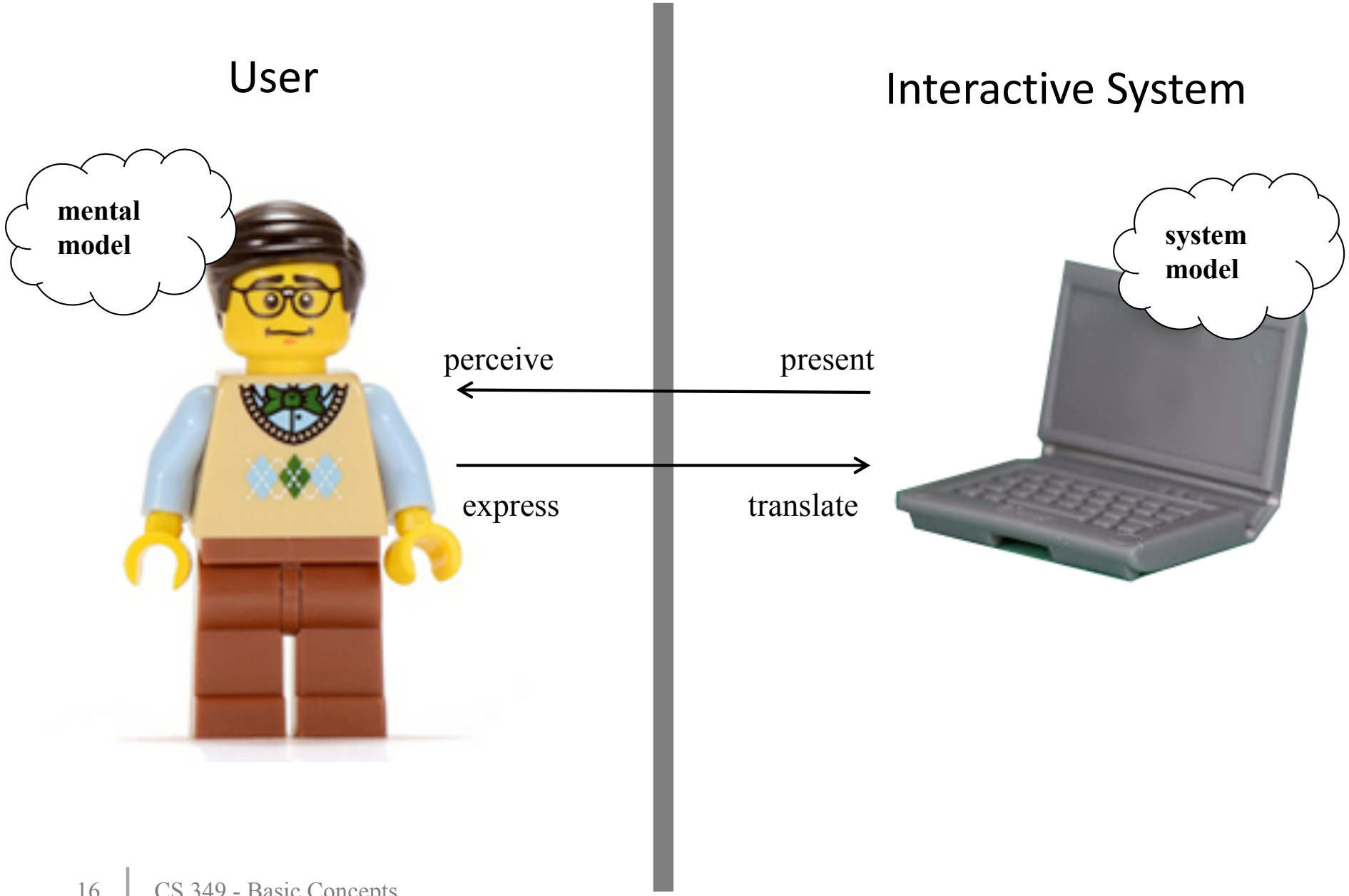
A user interface is the place where a person *expresses intention* to an artifact, and the artifact *presents feedback* to the person.

Interactive System Architecture

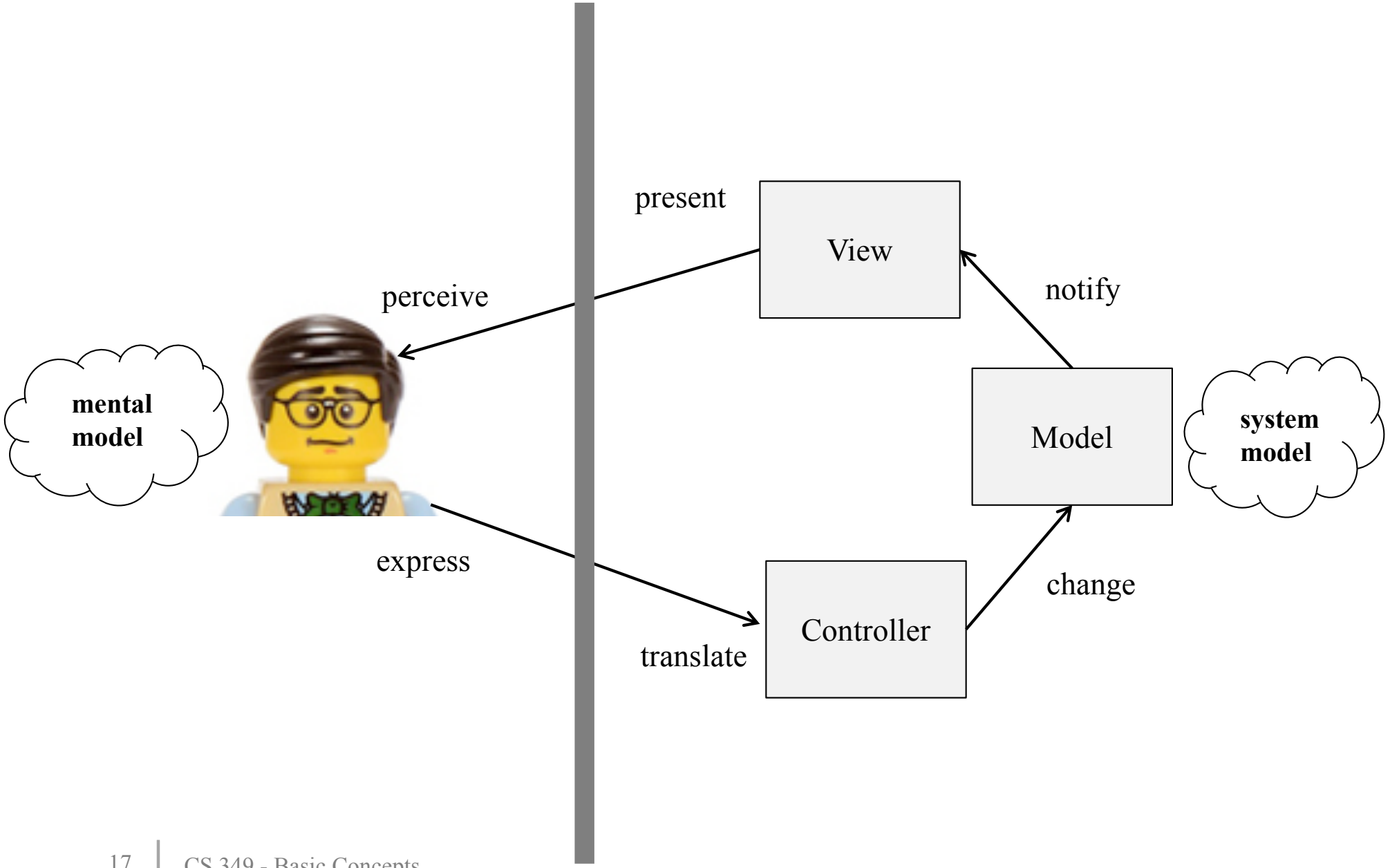
Interactive System Architecture



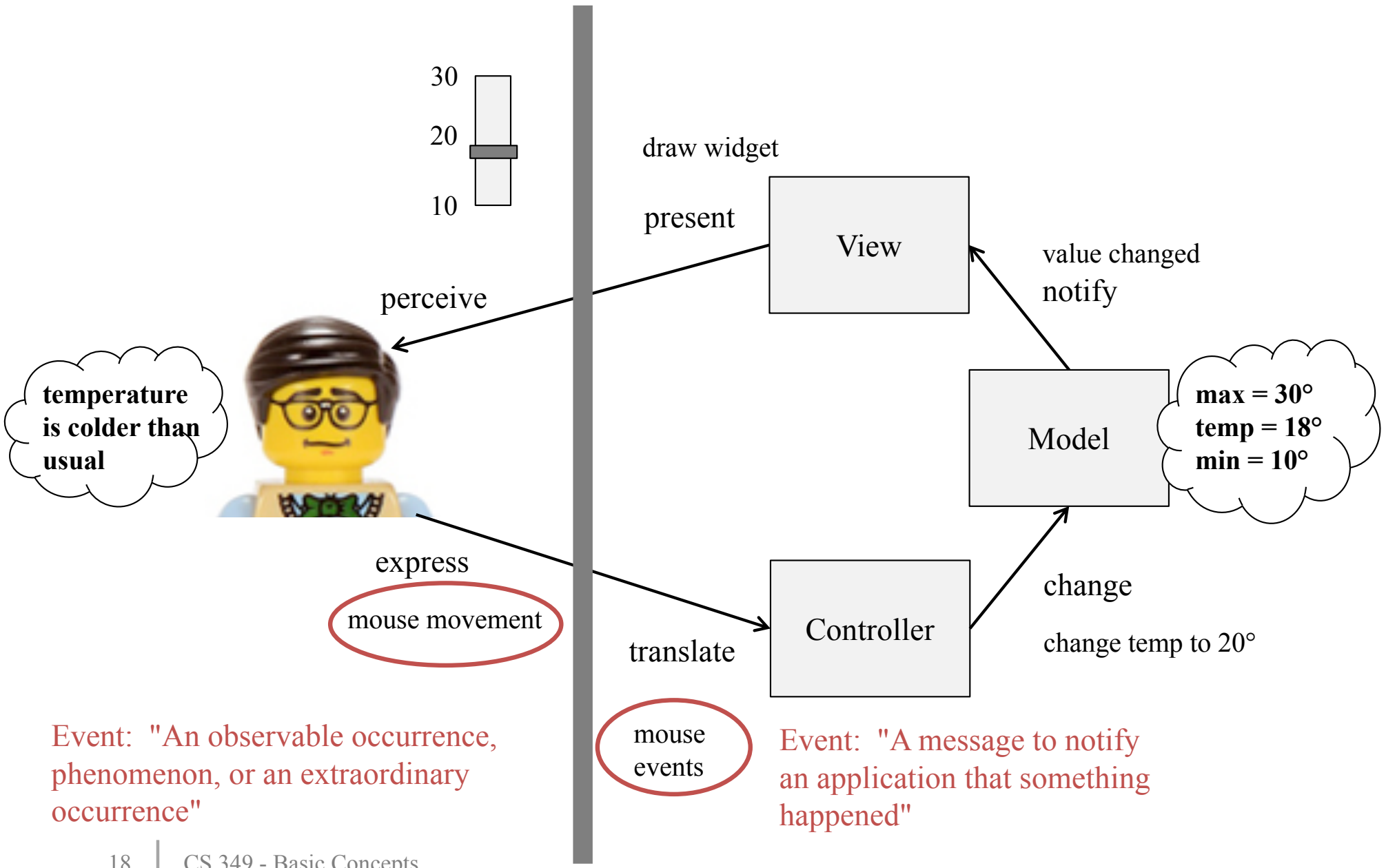
Interactive System Architecture



Model-View-Controller (MVC)



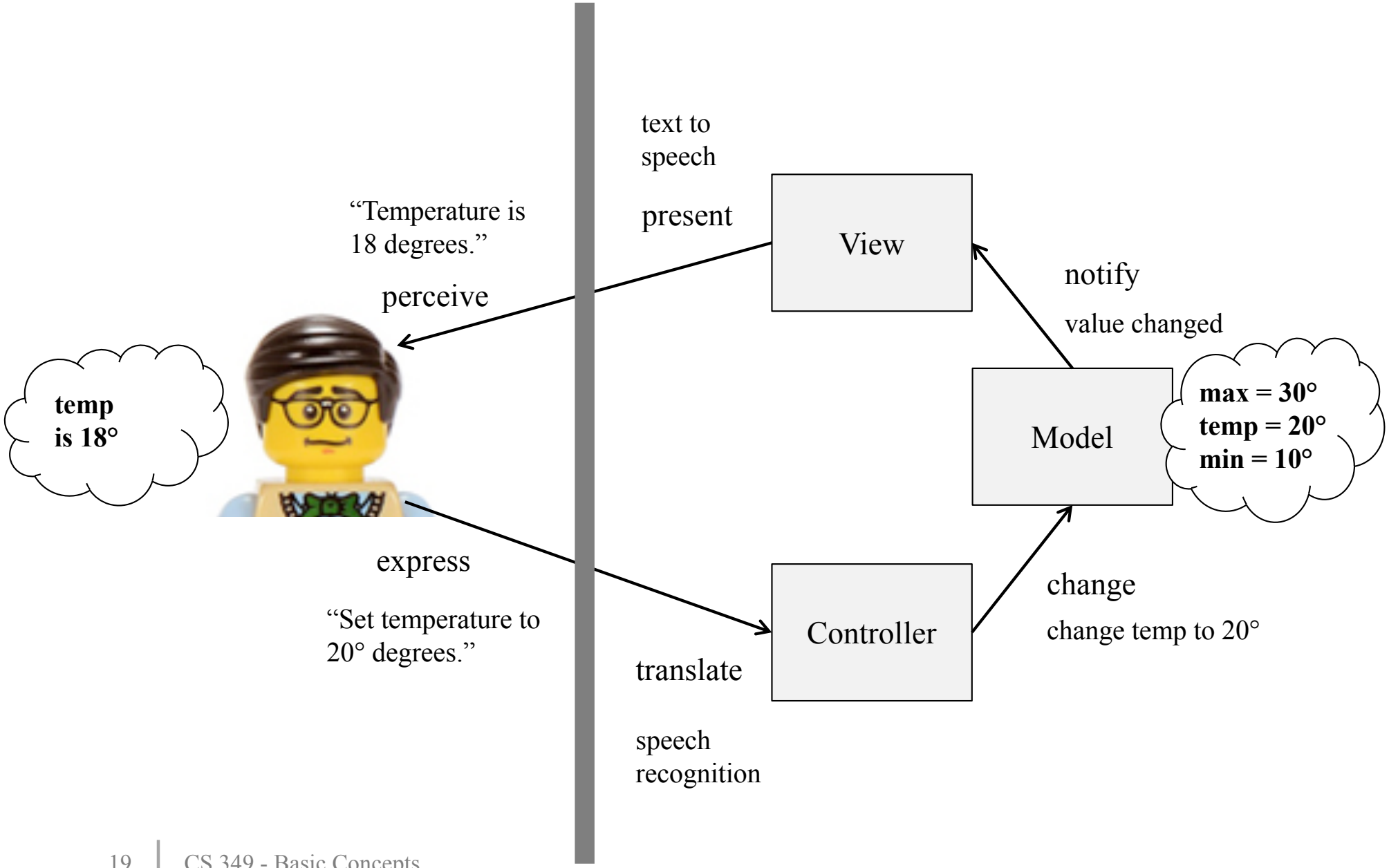
Graphical Temperature Control



Event: "An observable occurrence, phenomenon, or an extraordinary occurrence"

Event: "A message to notify an application that something happened"

Speech Temperature Control

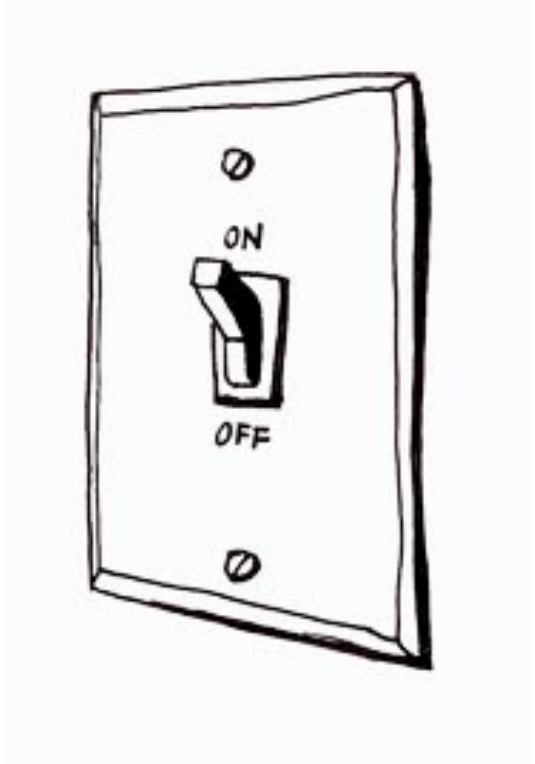


Interface vs. Interaction

- What is the difference between an *interface* and *interaction*?
- **Interface** refers to the external presentation to the user
  - Controls (what you can manipulate to communicate intent)
  - Visual, physical, auditory presentation (what the program uses to communicate its response)
- **Interaction** is used to connote behavior: The actions the user must invoke to perform a task and the corresponding responses
  - Interaction is action and dialog
  - Unfolds over time

Interface and Interaction Design

- What is the interface?
- What is the interaction?



- Why is interaction design so hard?

Interaction Design

- Challenging because of variability in users and tasks
  - Varying levels of expertise among users
  - Often a range of tasks will performed with the same tool (i.e. tools can be generic) – can you anticipate all uses and scenarios?
- No one “right way” to design an interface; interfaces can always be improved.
- Pushing technology “forward” requires us to rethink interaction.
  - Emergence of UX as a discipline

Why Study Interaction? Empowering People

- Well designed interfaces empower people to do things they couldn't otherwise do
  - Desktop publishing, grassroots journalism (blogs), movie production, music production, image editing, assistive technologies…
- Interaction is the key to enabling new technologies
  - Multi-touch and gestures on smartphones
  - Voice interfaces for cars, watches
  - Touch screens on tablets, notebooks
- A well designed tool can change the world
  - The web browser, Linux, original Napster, the spreadsheet, email, instant messaging…

Syllabus

What to expect

How to be successful in this course!

Next steps

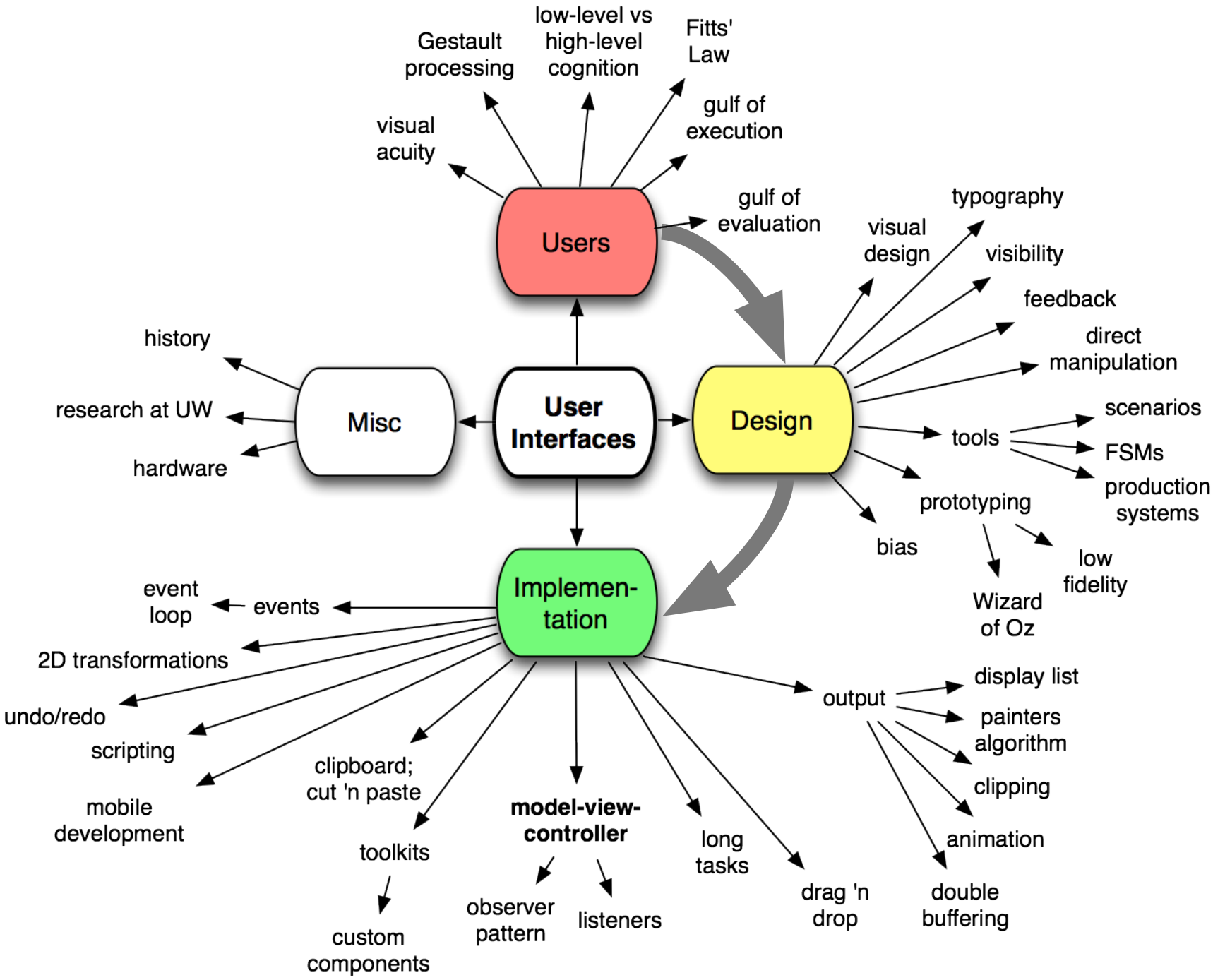
CS 349 Objectives

The goal of CS 349 is to teach you:

1) how to **design, implement, and evaluate** user interfaces
   - provide the foundational knowledge for building highly interactive, usable desktop, web and mobile applications
   - illustrate the underlying architecture of modern GUI toolkits
   - teach strategies applicable across a range of interface problems
   - teach essential design tools, techniques, and processes

2) ways to **understand users**
   - physical and cognitive abilities of users
   - visual design principles

User-centered design is covered more extensively in CS 449.


Gestault processing
low-level vs high-level cognition
Fitts' Law
visual acuity
gulf of execution
gulf of evaluation
Users
typography
visual design
visibility
feedback
direct manipulation
history
research at UW
hardware
Misc
User Interfaces
Design
scenarios
tools
FSMs
production systems
prototyping
bias
low fidelity
Wizard of Oz
event loop
events
Implemen-tation
2D transformations
undo/redo
scripting
mobile development
clipboard; cut 'n paste
toolkits
custom components
model-view-controller
observer pattern
listeners
long tasks
drag 'n drop
output
display list
painters algorithm
clipping
animation
double buffering

List of Topics

- Syllabus (introduction, motivation), Basic Concepts (user interfaces, interaction)
- Multiple windows (BWS, Windows Mgr, X examples)
- Java: advanced (interfaces, listeners, canvas, drawing)
- Events (event capture, event loop, dispatch) w. Java dispatch examples
- 2D graphics (shape models, mouse interaction, selection)
- MVC (rationale, demos, widgets)
- Widgets (physical vs. logical input, consistency, toolkits)
- Layout (Dynamic/Responsive, Layout Managers, Composite/Strategy DP)
- Visual perception (temporal/spatial resolution, color, displays)
- Users: visual design (gestalt principles, structure)
- Responsiveness (perception, UI feedback, Java UI threads, Web async calls)
- Undo/Redo; Cut/Paste
- Design principles (useful vs. usable, Norman, gulf evaluation/execution, metaphor, feedback)
- Design process (UCD, prototyping)
- History (interaction up to WIMP interfaces)
- Touch interfaces (Input, Interaction problems, DM on WIMP/touch, gestural, mobile)
- Touchless interfaces (gestural/speech design and problems)
- Android: advanced (intents, activities, mvc, graphics)
- Users: perception and cognition (cognetics, memory, bias)
- Accessibility (impairments, curb-cut, legal obligations)
- Input (types of input, keyboards, positional input, absolute/relative, direct/indirect)
- Input performance (KLM, Fitt's Law, Steering Law)
- HCI research, methodology

CS349 People

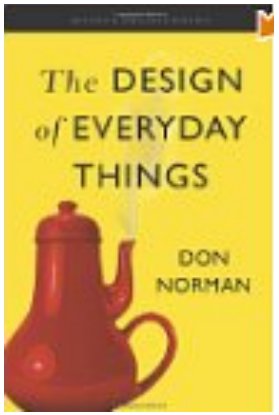
- Instructors:
  - Jeff Avery & Alex Williams
- Instructional Support Coordinator (ISC):
  - Caroline Kierstead

- Instructional Assistants (IA):
  - Shaishav Siddhpuria
  - Terence Dickson
- Teaching Assistants (TAs):
  - Bahareh Sarrafzadeh
  - Dallas Fraser
  - Jeremy Hartmann
  - Lisa Elkin
  - Qi Feng (Edmund) Liu
  - Will Callaghan

Website, Schedule, Textbook

- Web Site: www.student.cs.uwaterloo.ca/~cs349/

- Schedule on website
  - lecture topics, slides, sample code
  - assignment & midterm dates
- Slides posted before class
  - Lectures may include more than what is covered in the slides
- Textbooks - useful but not required!
  - *Building Interactive Systems*, Dan R. Olsen Jr.
  - *The Design of Everyday Things*, Donald A. Norman
  - *Designing with the Mind in Mind*, Jeff Johnson



Lectures

Days	Time	Room	
▪ Mon, Wed, Fri	10:30-11:20	MC 4040	Jeff
▪ Mon, Wed, Fri	1:30-2:20	MC 4040	Jeff
▪ Mon, Wed, Fri	2:30-3:20	MC 4040	Jeff
▪ Mon, Wed, Fri	3:30-4:20	MC 4040	Alex

We'll attempt to keep all sections in-sync; if you miss your scheduled lecture, you have other chances to catch it!

During lectures:

- Participate by asking and answering questions
- Try and limit computer use for anything other than taking notes.

Assignments

Assignments meant to provide meaningful, engaging experiences in constructing interfaces… while giving you the opportunity to create applications you will want to share with others.

- Four assignments, spaced (roughly) 3 weeks apart.
  - You will build desktop applications using C++ and Java.
  - You are expected to learn Git and Java on your own time!
  - A "getting started" tutorial will be offered next week.
- There's lots of room for creativity in assignments
  - Typically have a component for going above and beyond the spec
- Assignments require significant time coding
  - Do not underestimate the time it takes to code interactive applications that are intuitive and easy-to-use. Start early!
- Individual NOT group assignments

What You'll Need for Assignments

- A1: X Windows
  - C++ and Xlib (XWindows)
  - Can be done on your laptop (Linux, Mac) or in the student environment.
  - Required: editor, g++ compiler, xlib installed, git client
- A2-A4: Java
  - Java 8 SE
  - Can be done on your laptop (Linux, Mac, Windows) or in the student environment.
  - Required: editor, Java 8 compiler/runtime, git client
  - Recommended: IntelliJ IDE

You're expected to learn Java and Git on your own time!

Assignment Policies

- Due dates: 5:00pm on the date in schedule
  - Late assignments are NOT accepted, except for valid medical reasons (with a doctors note provided to Caroline).
- Assignments submissions are through your private repo on Git
  - Repos will be setup for you Monday of the second week of the term
  - http://git.uwaterloo.ca
- Assignments are your individual work
  - You may use examples provided in class and on the course website
  - You should NOT be doing Internet searches for specific solutions.
- You should NOT post your solutions online.
  - Google indexes GitHub (i.e. anyone can easily find your repo).
  - If someone finds your assignment on GitHub and uses your code, it's an academic offense for BOTH of you.

Getting Help Outside of Lectures

- Office Hours
  - Lecture and assignment help
  - Details posted on course website
- Piazza for Q&A
  - Everyone should sign up! Please use a meaningful name
  - Staff will monitor (best-effort response, no time-guarantees)
  - When posting
    - Search before you post!
    - Use a meaningful title for every post
    - Answer questions, but don't be too explicit
    - Build one collaborative answer!

Grading

- Assignments: 40%
  - 10%, 10%, 10%, 10%
- Midterm: 20%
  - Wed Feb 15 @ 7:00 PM – 8:50 PM (location TBD)
- Final: 40%
  - Scheduled by the Registrar's Office

- To pass the course, you must pass the weighted exam average and the weighted assignment average
- If you fail the assignments, you fail the course, regardless of how well you do on the exams.

Next Steps

This week

- Explore the web site:
  http://student.cs.uwaterloo.ca/~cs349
- Get signed up for Piazza:
  http://www.piazza.com/uwaterloo.ca/winter2017/cs349/home
- Login to the student environment
  – ssh username@linux.student.cs.uwaterloo.ca
- Login to Git:
  http://git.uwaterloo.ca
  - Logging in creates your account – required for us to create your repos!
  - Repositories will be created later this week (we'll post details on Piazza)

Next week

- Install Java 8 JDK, Git client (graphical, or command line)

Questions?