

HCI Summary

- Introduction to Human-Computer Interaction
- Useful vs. Useable
- Graphical User Interface (GUI)
- Java GUI (Swing)

April 1, 2009

Slides by Mark Hancock

1

By the end of this lecture, you will be able to describe what **human-computer interaction** is.

You will also be able to distinguish between **usable** and **useful** programs.

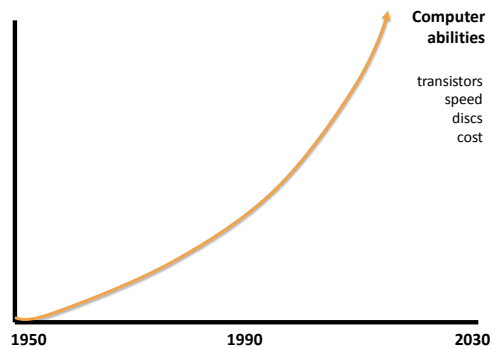
You will also be able to describe what an **observer** design pattern is.

April 1, 2009

Slides by Mark Hancock

2

Moore's Law

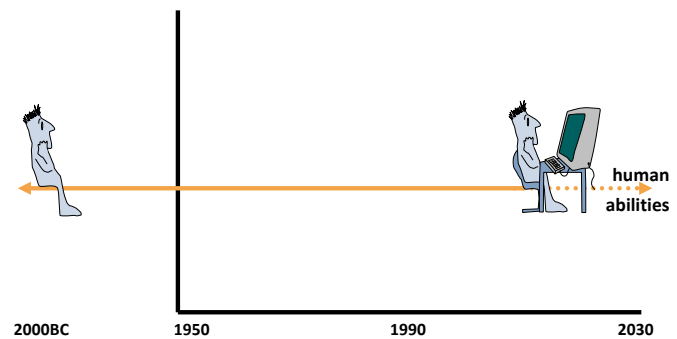


April 1, 2009

Slides by Mark Hancock
(Slide idea by Bill Buxton)

3

Psychology

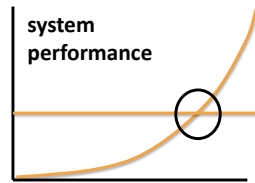


April 1, 2009

Slides by Mark Hancock
(Slide idea by Bill Buxton)

4

Where is the bottleneck?



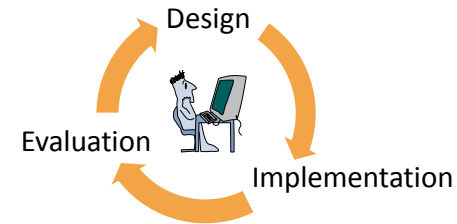
April 1, 2009

Slides by Mark Hancock
(Slide idea by Bill Buxton)

5

Human Computer Interaction

- A discipline concerned with the



- of interactive computing systems for human use

April 1, 2009

Slides by Mark Hancock
(adapted from notes by Saul Greenberg)

6

Discussion: contrast this to our discussion about knowledge and understanding.

Where do the problems we analyse come from?

April 1, 2009

Slides by Mark Hancock

7

Discussion: How would you evaluate a program that you created?

What would you evaluate?

April 1, 2009

Slides by Mark Hancock

8

Useful vs. Useable

- Useful:
 - refers to function or purpose
- Useable:
 - refers to human ability

April 1, 2009

Slides by Mark Hancock

9

Exercise: What's an example of something that is **useful**, but not **useable**.

April 1, 2009

Slides by Mark Hancock

10

Exercise: What's an example of something that is **useable**, but not **useful**.

April 1, 2009

Slides by Mark Hancock

11

Question: which of these two qualities is important when you make a program?

April 1, 2009

Slides by Mark Hancock

12

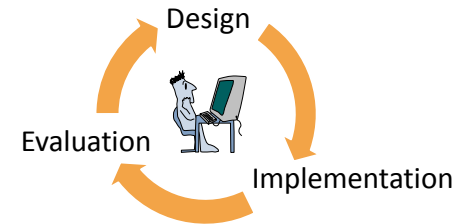
Challenge: How do you make your program both useful and useable?

April 1, 2009

Slides by Mark Hancock

13

Iterative Design



April 1, 2009

Slides by Mark Hancock
(adapted from notes by Saul Greenberg)

14

Discussion: How would using an iterative design cycle affect how you write your code?

What would make it easier to iterate?

April 1, 2009

Slides by Mark Hancock

15

Graphical User Interface (GUI)

- Most programs you've written so far use the command line
- Most programming languages have a way to create GUIs (sometimes through a separate library)

April 1, 2009

Slides by Mark Hancock

16

Discussion: If you were to design a program that uses a GUI, how would it differ from what you've written so far?

How would you handle a mouse click? or when someone presses a button?

April 1, 2009

Slides by Mark Hancock

17

Observer Design Pattern

"The **observer pattern** ... is a software design pattern in which an object, called the **subject**, maintains a list of its dependents, called **observers**, and **notifies them** automatically of any **state changes**, usually by calling one of their methods. It is mainly used to implement distributed event handling systems."

"Observer Pattern," *Wikipedia: The Free Encyclopedia*

Often referred to as: The **Event Model**

April 1, 2009

Slides by Mark Hancock

18

Java Swing

- Tutorial:
<http://java.sun.com/docs/books/tutorial/uiswing/>
- Use **inheritance** to create buttons, menus, tabs, etc.
- Use an **event model** to handle mouse clicks, keyboard/button presses, etc.

April 1, 2009

Slides by Mark Hancock

19

Example: GUI in Java

```
import javax.swing.*;

public class SimpleGUIProgram extends JFrame
{
    public SimpleGUIProgram()
    {
        super("A GUI!");
    }

    public static void main(String[] args)
    {
        SimpleGUIProgram program = new SimpleGUIProgram();
        program.setVisible(true);
    }
}
```

April 1, 2009

Slides by Mark Hancock

20

Example (cont'd)

```
public SimpleGUIProgram()
{
    super("A GUI!");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JButton button = new JButton("Okay");
    button.addActionListener(this);

    JTextArea textArea = new JTextArea(70, 100);

    JScrollPane scrollPane = new JScrollPane(textArea);
    scrollPane.setPreferredSize(new Dimension(640, 480));

    JPanel mainArea = new JPanel();
    mainArea.add(scrollPane);
    mainArea.add(button);

    setContentPane(mainArea);
    pack();
}
```

April 1, 2009

Slides by Mark Hancock

21

Example (cont'd)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SimpleGUIProgram extends JFrame
                                implements ActionListener
{
    // ...

    public void actionPerformed(ActionEvent event)
    {
        JOptionPane.showMessageDialog(this,
                                      "You pressed okay!");
    }

    // ...
}
```

April 1, 2009

Slides by Mark Hancock

22

Demo

April 1, 2009

Slides by Mark Hancock

23

Discussion: What kind of events would you have to consider in the demo you just saw?

April 1, 2009

Slides by Mark Hancock

24

HCI Summary

- Introduction to Human-Computer Interaction
- Useful vs. Useable
- Graphical User Interface (GUI)
- Java GUI (Swing)

April 1, 2009

Slides by Mark Hancock

25

Next Class

- I will be absent: Christopher Collins will provide the lecture
- Design Patterns (Part 2)

April 1, 2009

Slides by Mark Hancock

26