Representation II & Implementing Interfaces

Human Computer Interaction
CSCI 4620U| SOFE 4850U | CSCI 5540G
Dr. Christopher Collins

Acknowledgement: Parts of these lectures are based on material prepared by Ron Baecker, Ravin Balakrishnan, John Chattoe, Ilona Posner, Scott Klemmer, and Jeremy Bradbury.

Your location: Assessments › Assessment Manager › Reading Quiz 1 › Question Attempts › Democratization of Media › **Democratization of Media**

View Question Attempt

View Attempt 1 of 1

[redacted]

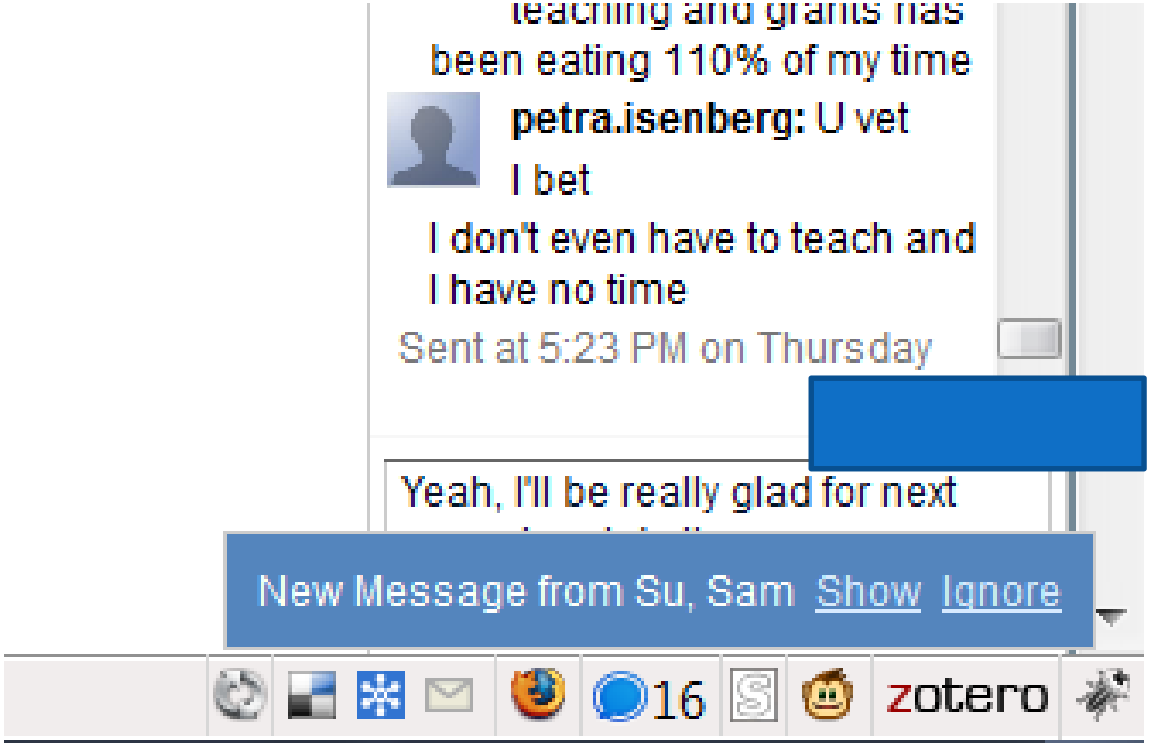
Democratization of Media

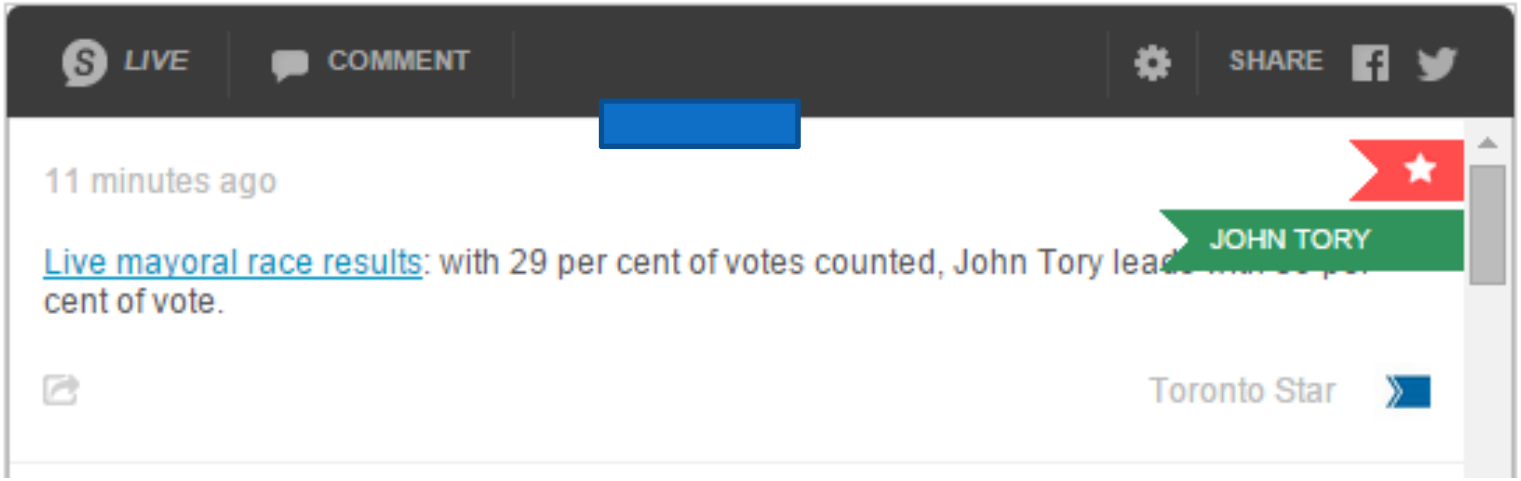
One of the biggest changes brought by digital technology is the "democratization of media". What is "democratization of media" and how do computers help?

Student Response: Media sharing is no longer a "dictatorship" and now anyone has the right to produce their own media and spread it via youtube, etc. This is called a democratization because it is similar to a democracy. Computers help in the spread and creation of this media.

Score: / 20

Comments for Student


teaching and grants has
been eating 110% of my time
petra.isenberg: U vet
I bet
I don't even have to teach and
I have no time
Sent at 5:23 PM on Thursday
Yeah, I'll be really glad for next
New Message from Su, Sam Show Ignore
16
zotero


LIVE
COMMENT
SHARE
11 minutes ago
JOHN TORY
Live mayoral race results: with 29 per cent of votes counted, John Tory lea
cent of vote.
Toronto Star

REVIEW

Last Week

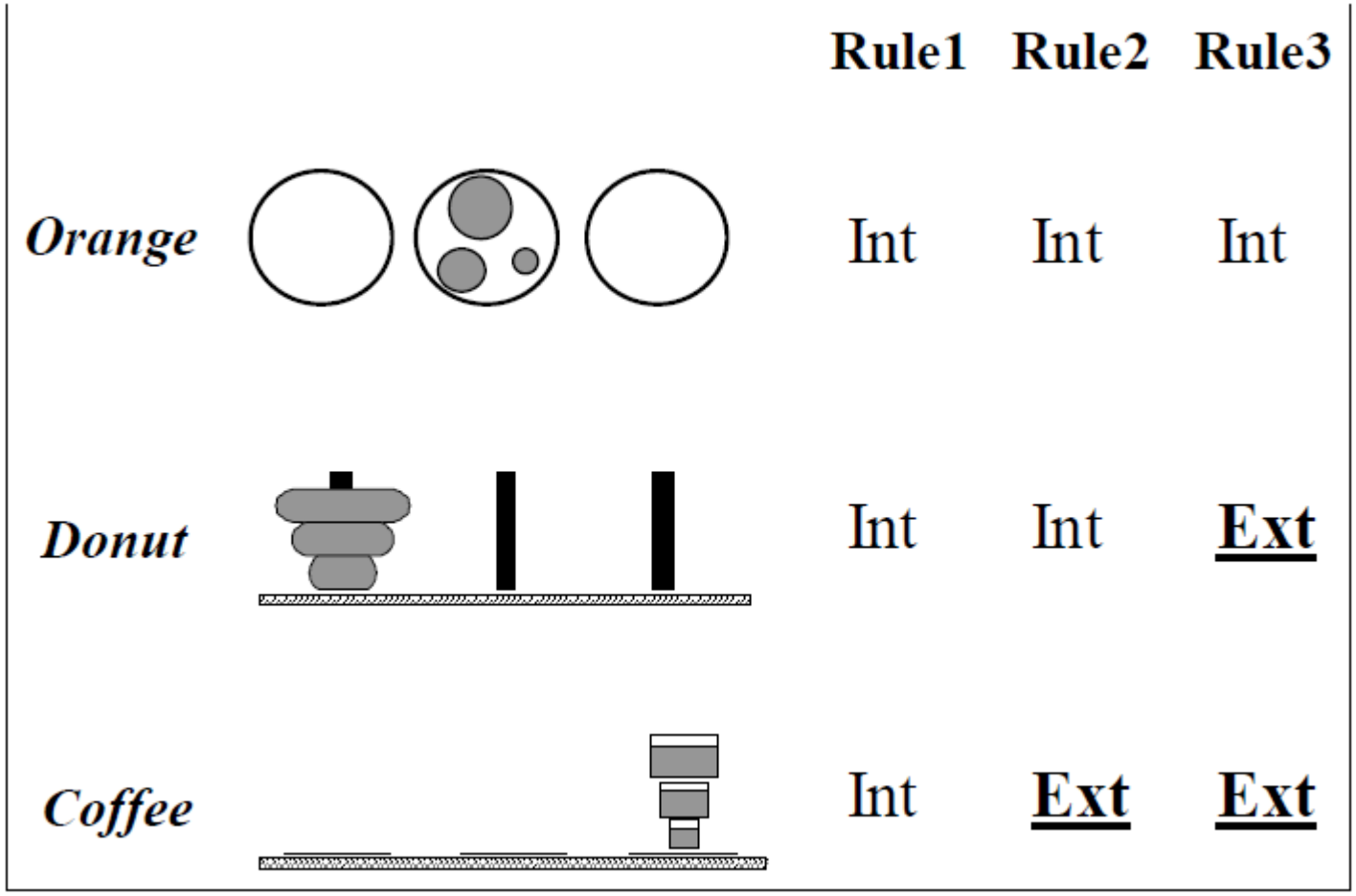
- Visual (Screen) Design
- Navigation Design
- Design for the Environment
- Representation
  - Problem isomorphs

Navigation Design

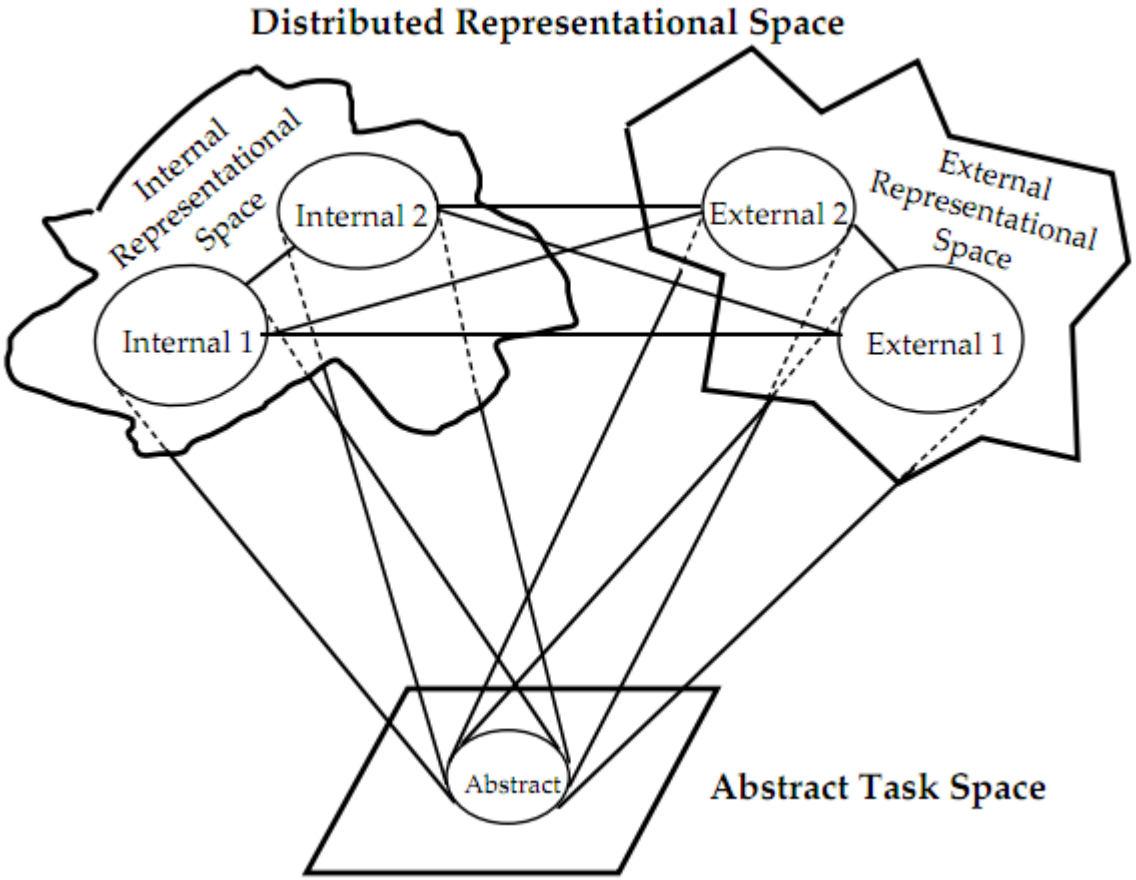
- Local structure refers to navigating from the current screen
- Global structure: how screens relate to one another
  - Hierarchy and network interface diagrams

Designing for the Environment

- Ensure applications conform to the standard look-and-feel established by the platform
- Ensure applications handle standard features and meet I/O standards
- Ensure applications work with other applications when needed



Jiajie Zhang and Donald A. Norman, Representations in Distributed Cognitive Tasks, *Cognitive Science*, 18, pp 87-122, 1994. doi=10.1.1.73.7799

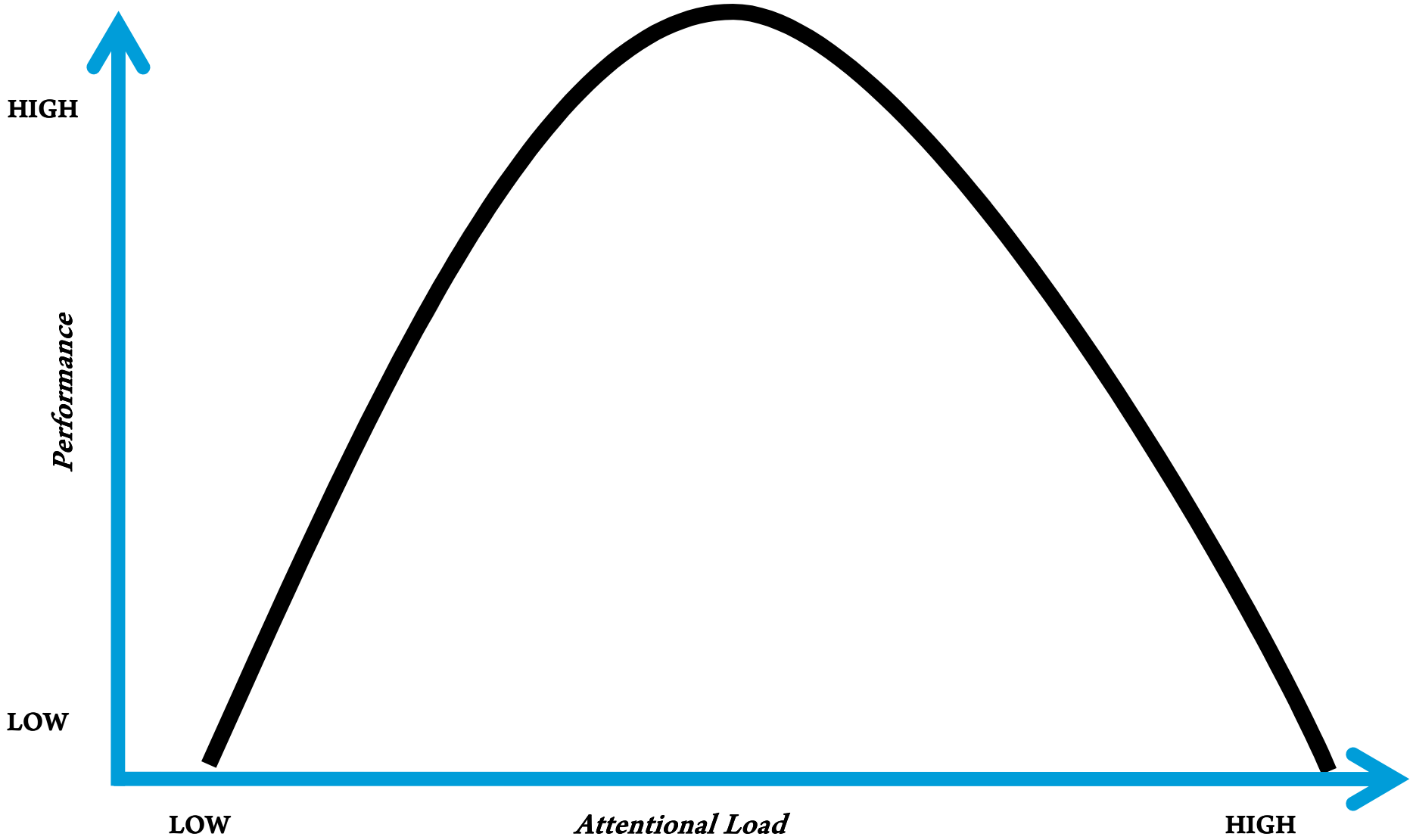


Jiajie Zhang and Donald A. Norman, Representations in Distributed Cognitive Tasks, *Cognitive Science*, 18, pp 87-122, 1994. doi=10.1.1.73.7799

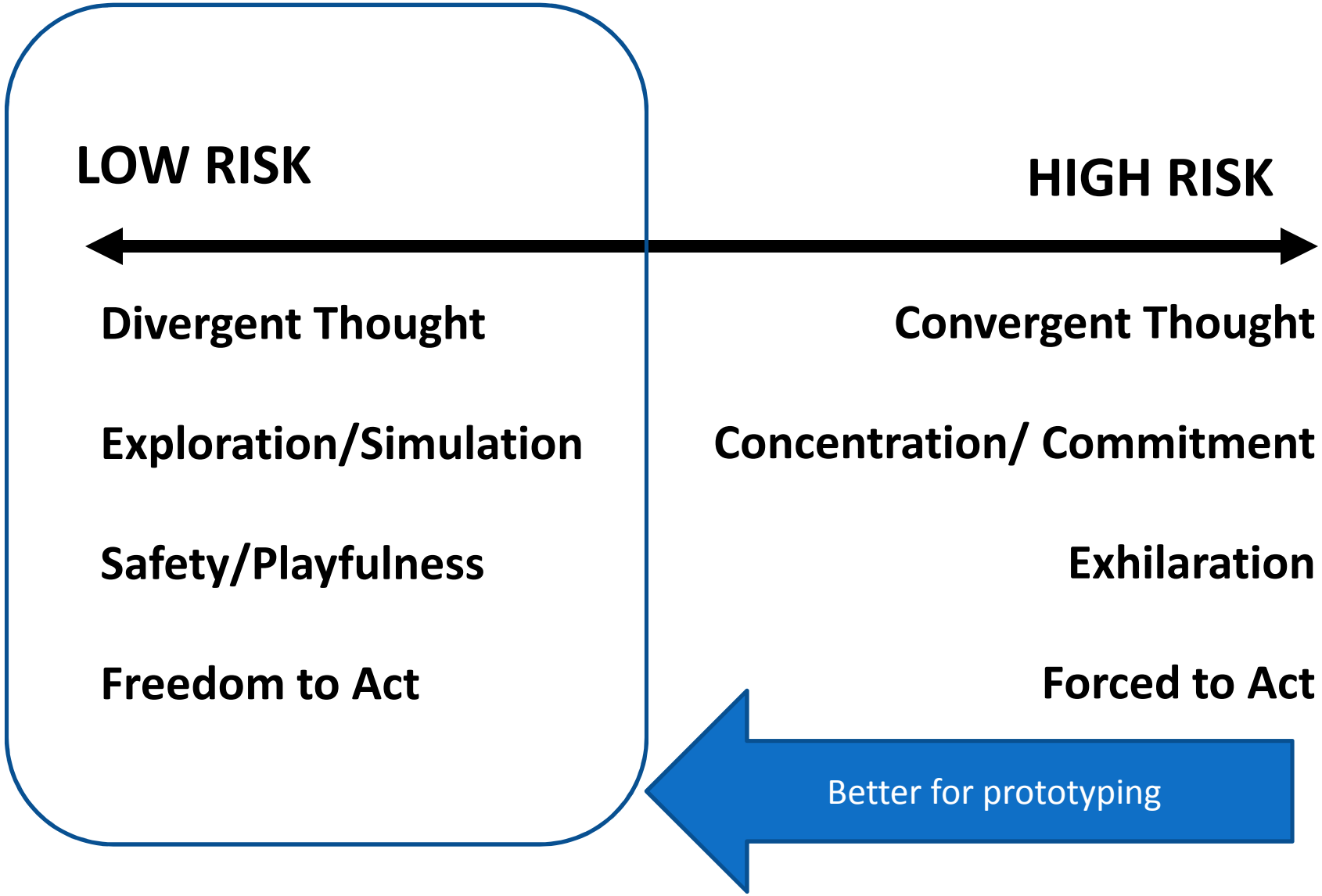
Solving a problem simply means representing it so as to make the solution transparent

(Simon, 1981)

Attention, Stress, and Risk

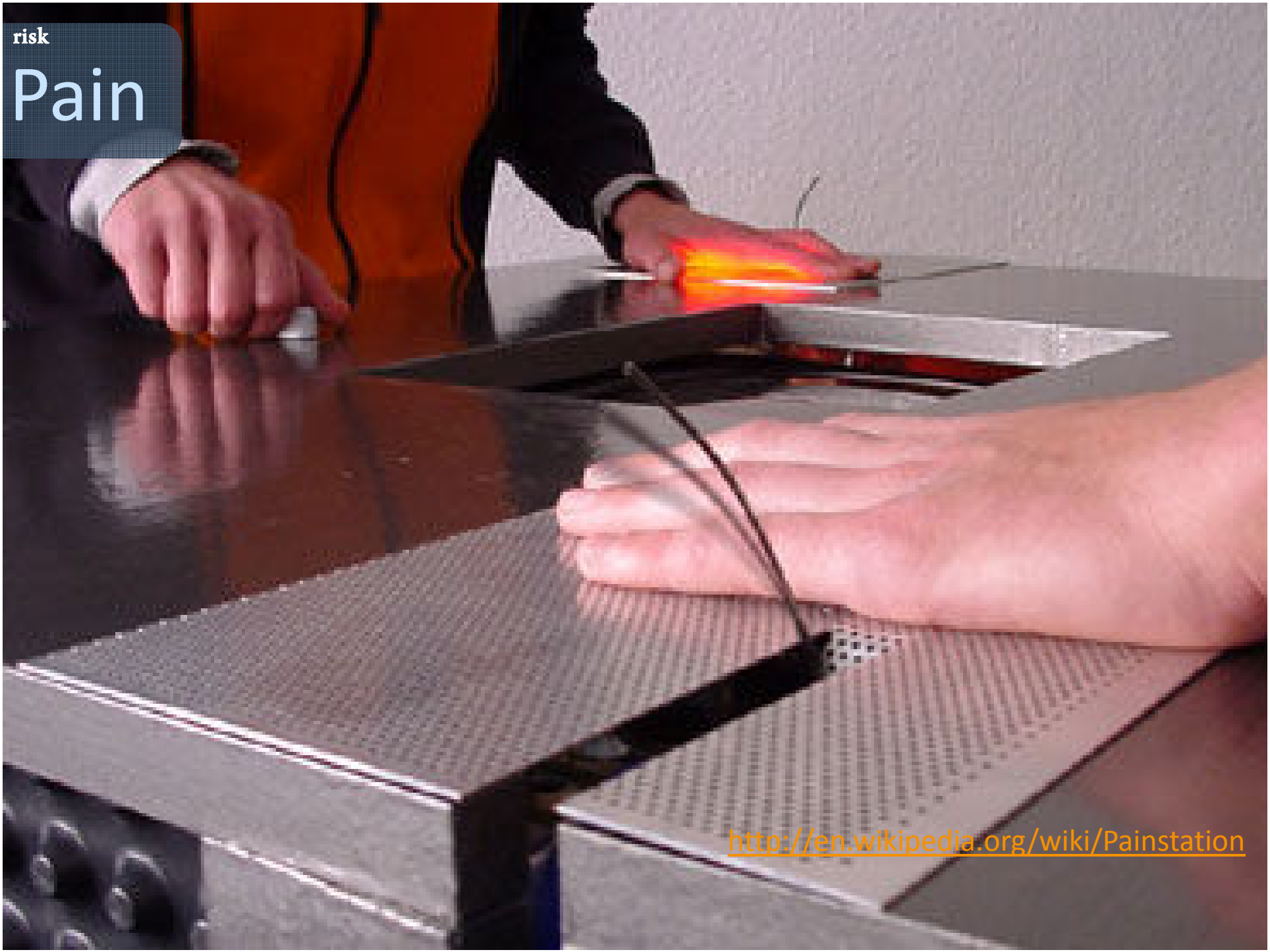


Tradeoffs


LOW RISK
HIGH RISK
Divergent Thought
Exploration/Simulation
Safety/Playfulness
Freedom to Act
Convergent Thought
Concentration/ Commitment
Exhilaration
Forced to Act
Better for prototyping

risk

Pain



Today

- Finish discussing representations
- Implementing interfaces

Designing for Cognitive Offloading

REPRESENTATION II

Representations

- A representation is
  - a formal system or mapping by which the information can be specified
  - a sign system in that it stands for something other than itself
- for example: the number thirty-four
    - decimal: 34,
    - binary: 100010,
    - roman: XXXIV
- different representations reveal different aspects of the information
    - decimal: counting & information about powers of 10,
    - binary: counting & information about powers of 2,
    - roman: counting
- presentation is not representation
    - how the representation is placed or organized on the screen
    - 34, 34, *34*

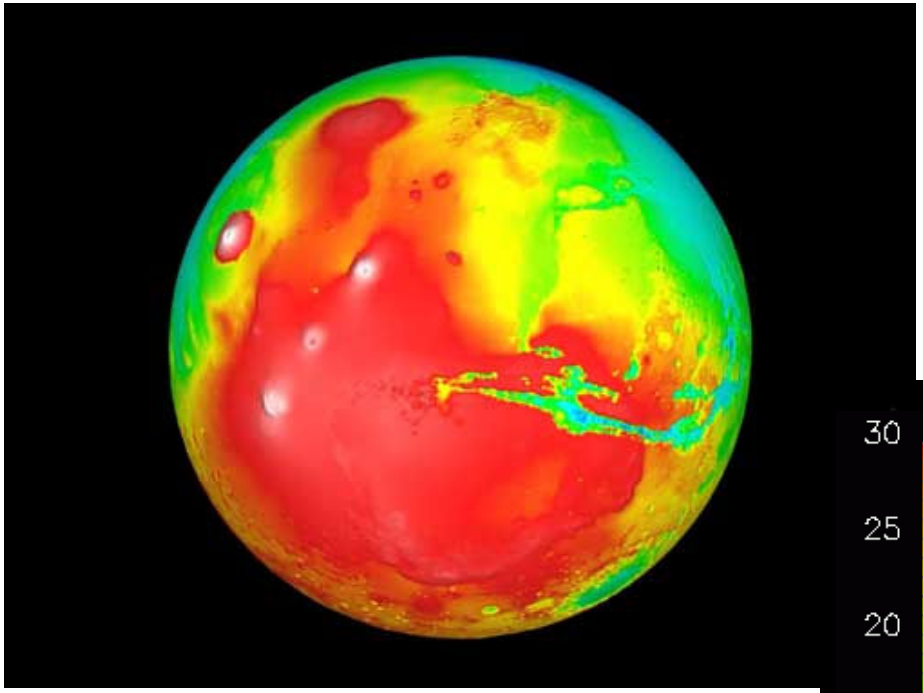
Representations

- Good representations
  - allow people to find relevant information
  - information may be present but hard to find
  - allow people to compute desired conclusions
  - computations may be difficult or “for free” depending on representations (long division using decimal vs. roman numerals)

Perceptual Principle

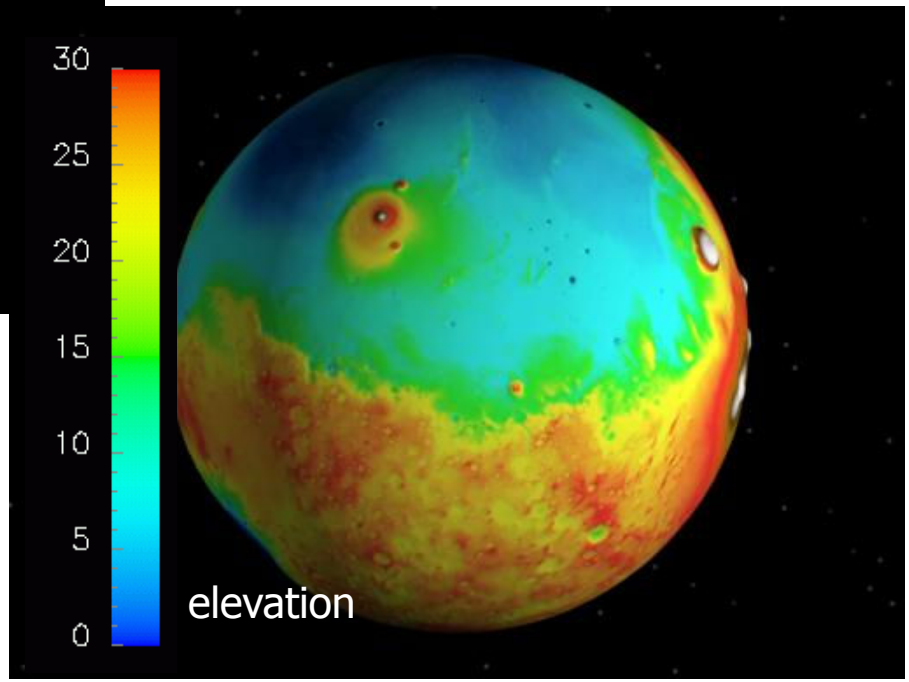
- Perceptual and spatial representations are more natural and therefore to be preferred over nonperceptual, nonspatial representations, but only if the mapping between the representation and what it stands for is natural -- analogous to the real perceptual and spatial environment

Avoiding Misinterpretation of Dimensions



3D Mars Globe

- If dimensions don't match expectations, provide a legend!



(Zoran Constantinescu)

Naturalness Principle

- External cognition is aided when the properties of the representation match the properties of the thing being represented

When do I take my drugs?

Note: 10 - 30% error rate in taking pills, same for pillbox organizers

Inderal -1 tablet 3 times a day
Lanoxin -1 tablet every a.m.
Carafate - 1 tablet before meals and at bedtime
Zantac - 1 tablet every 12 hours (twice a day)
Quinag - 1 tablet 4 times a day
Couma - 1 tablet a day

Organized by drug

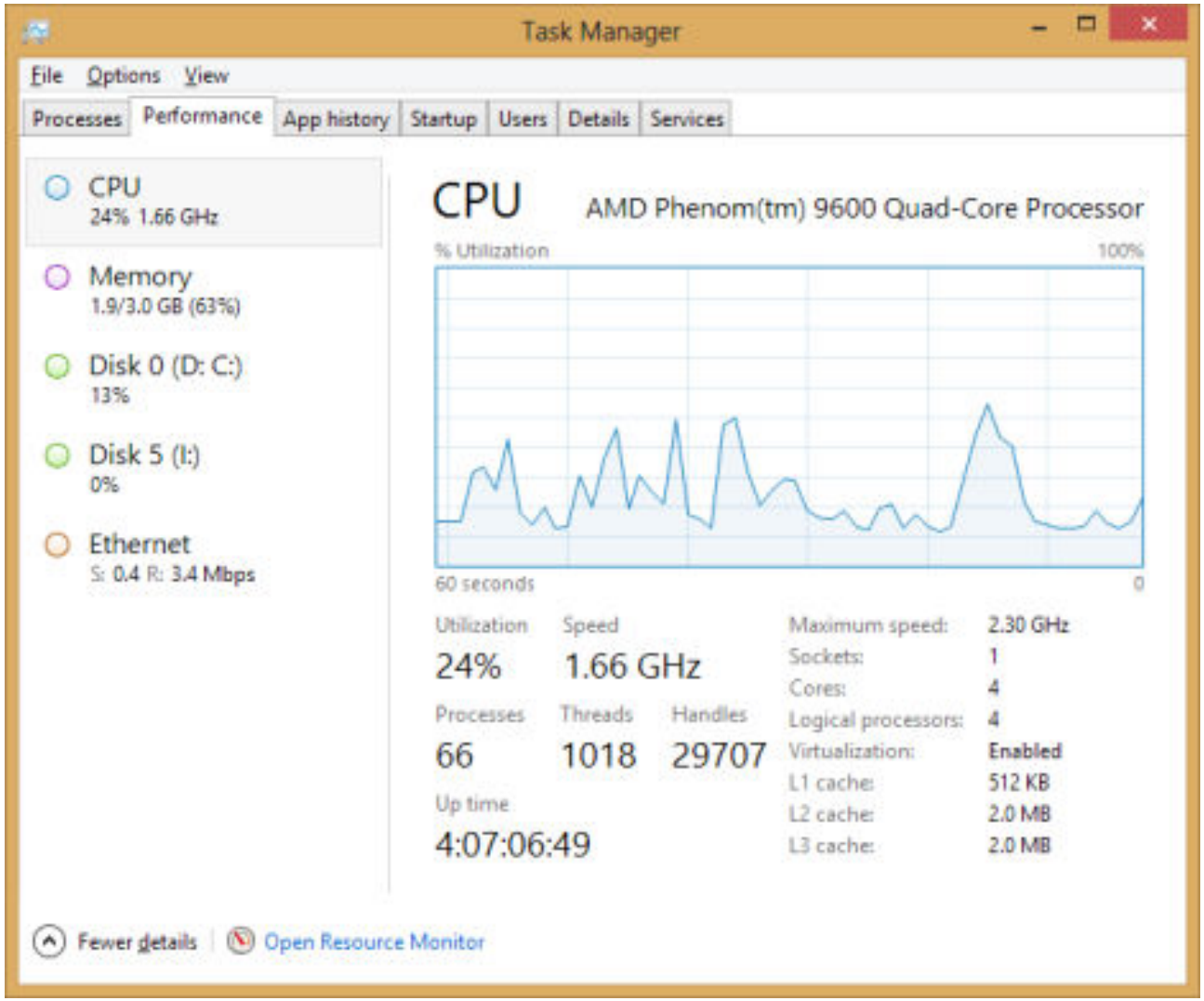
Breakfast	**Lunch**	**Dinner**	**Bedtime**
Lanoxin	Inderal	Inderal	Inderal
Inderal	Quinag	Quinag	Quinag
Quinag	Carafate	Carafate	Carafate
Carafate	Zantac		Zantac
			Couma

Organized by time of day

	Breakfast	Lunch	Dinner	Bedtime
Lanoxin	O			
Inderal	O	O	O	O
Quinag	O	O	O	O
Carafate	O	O	O	O
Zantac		O		O
Couma				O

Organized by both time of day and by drug

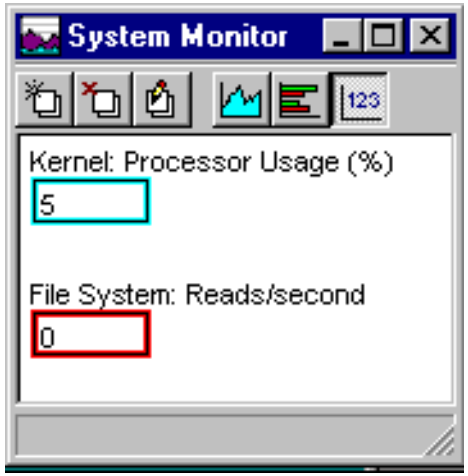
Best Representation Depends on the Task



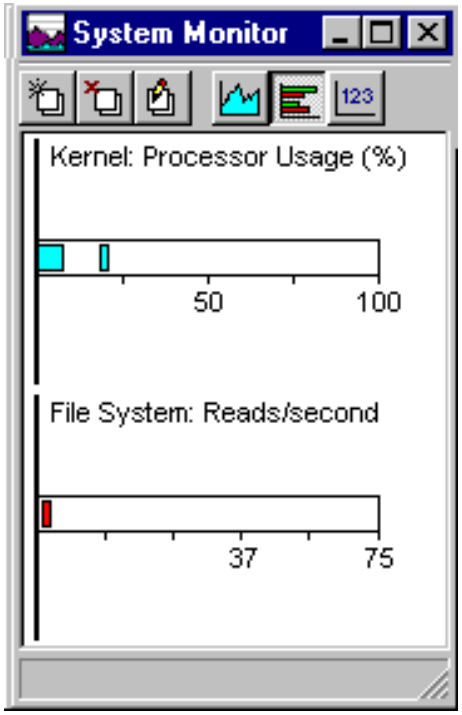
Temporal variation: visual

Precise CPU utilization: numeric

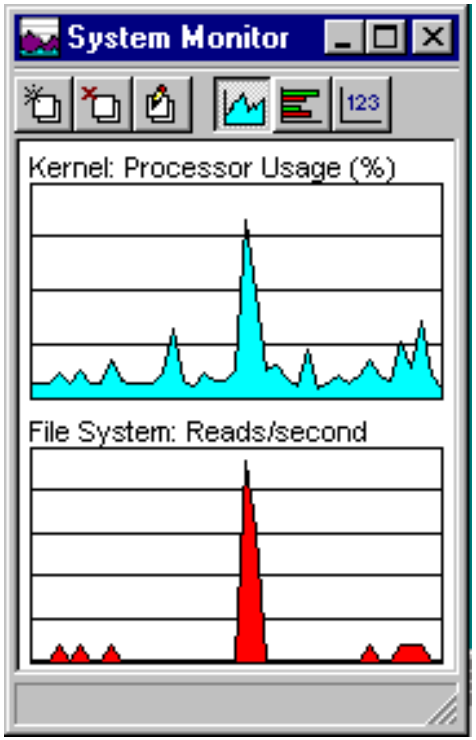
Best Representation Depends on the Task



What is precise value?

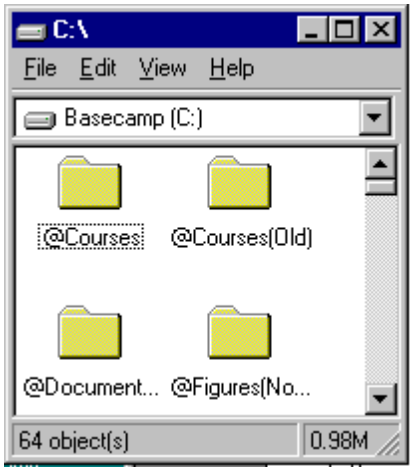


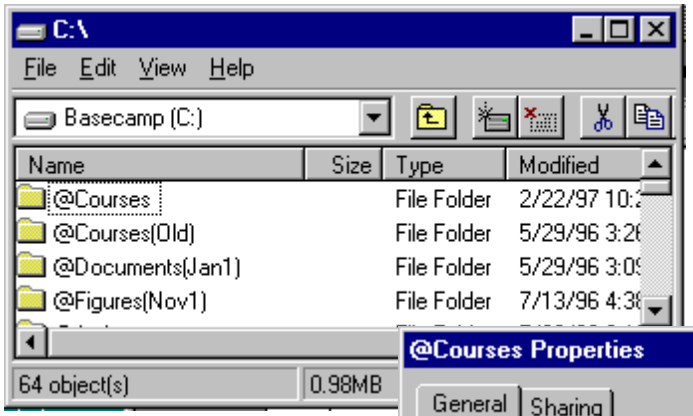
How does the performance now compared to its peak?



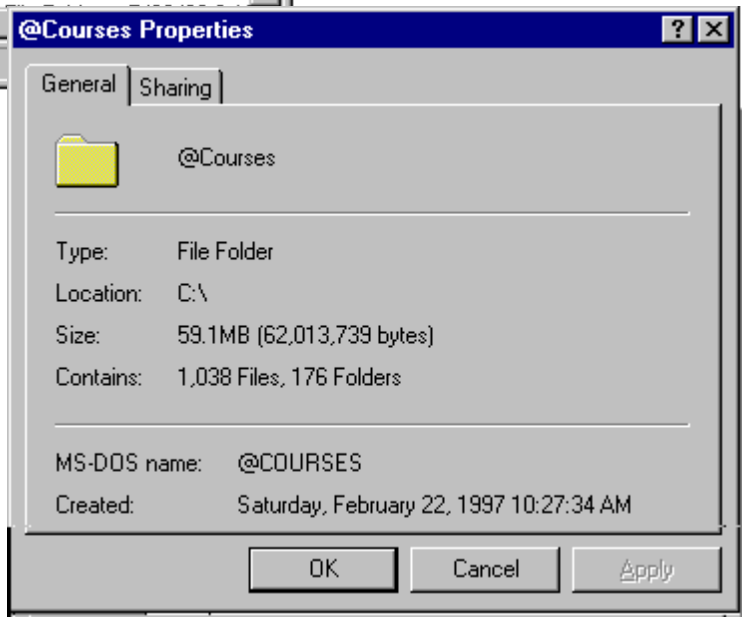
How does performance change over time?

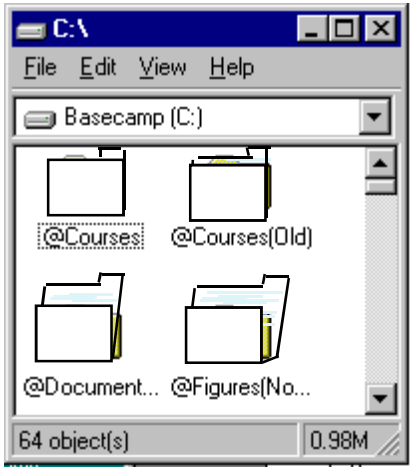
Which Folder is the Largest?



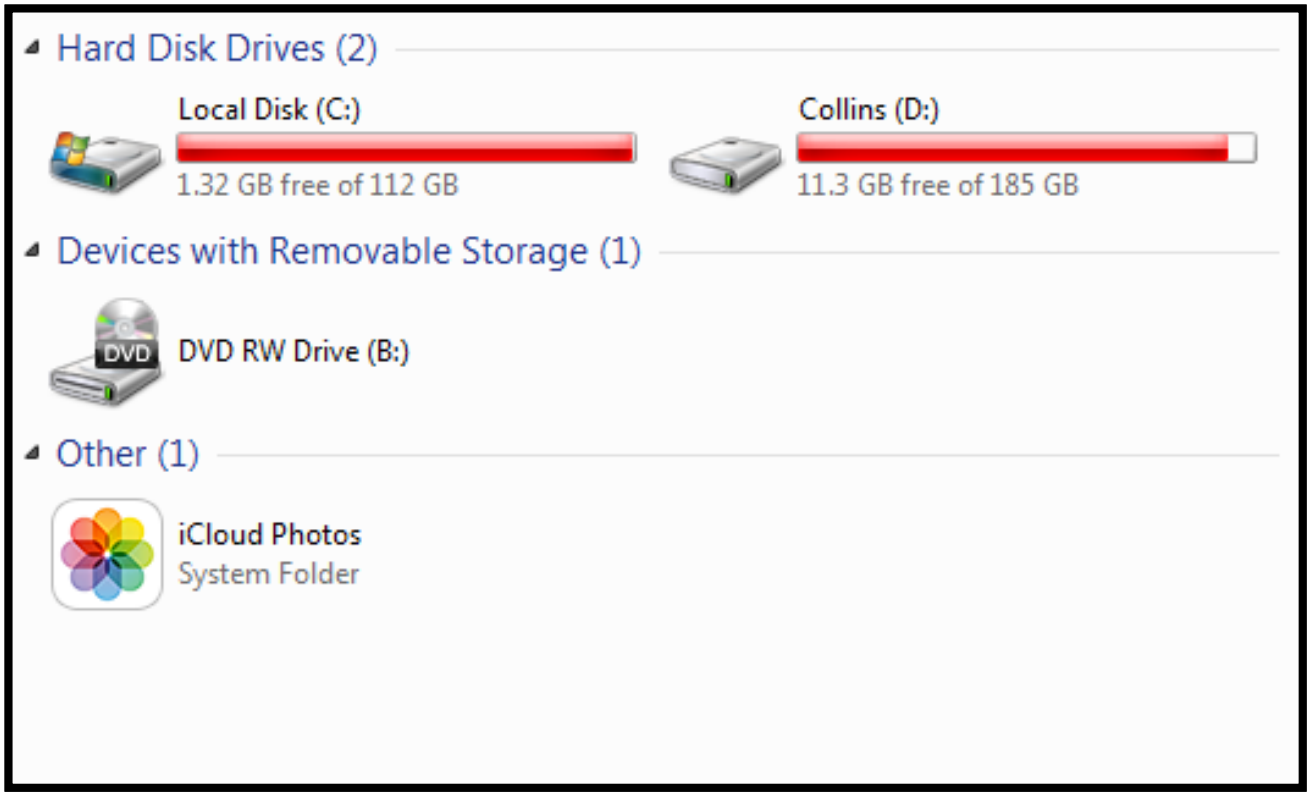


right menu
+ properties



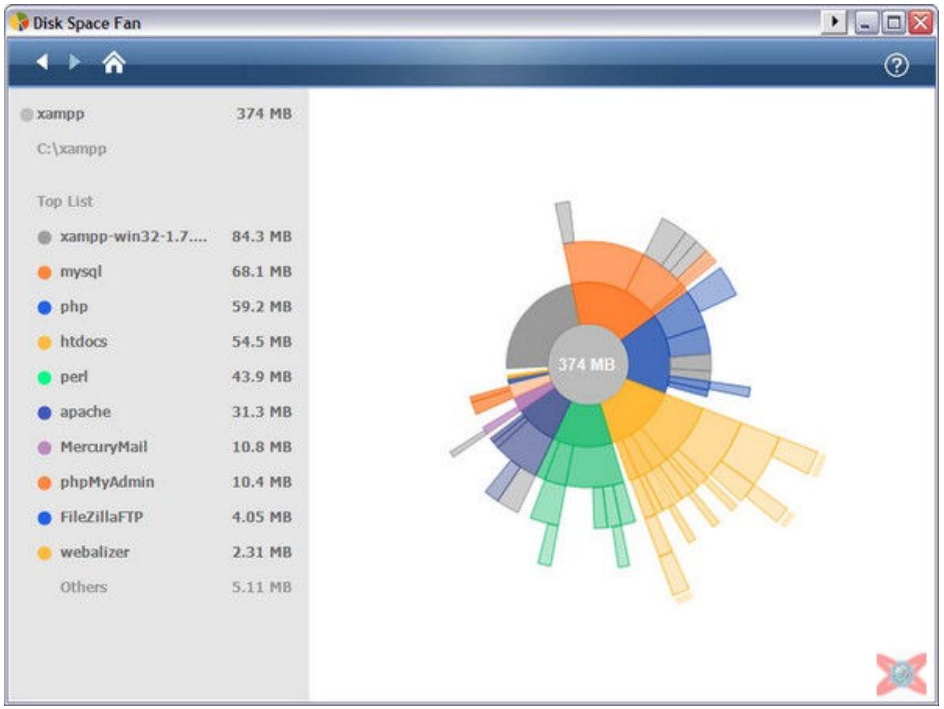


Disk Utilization



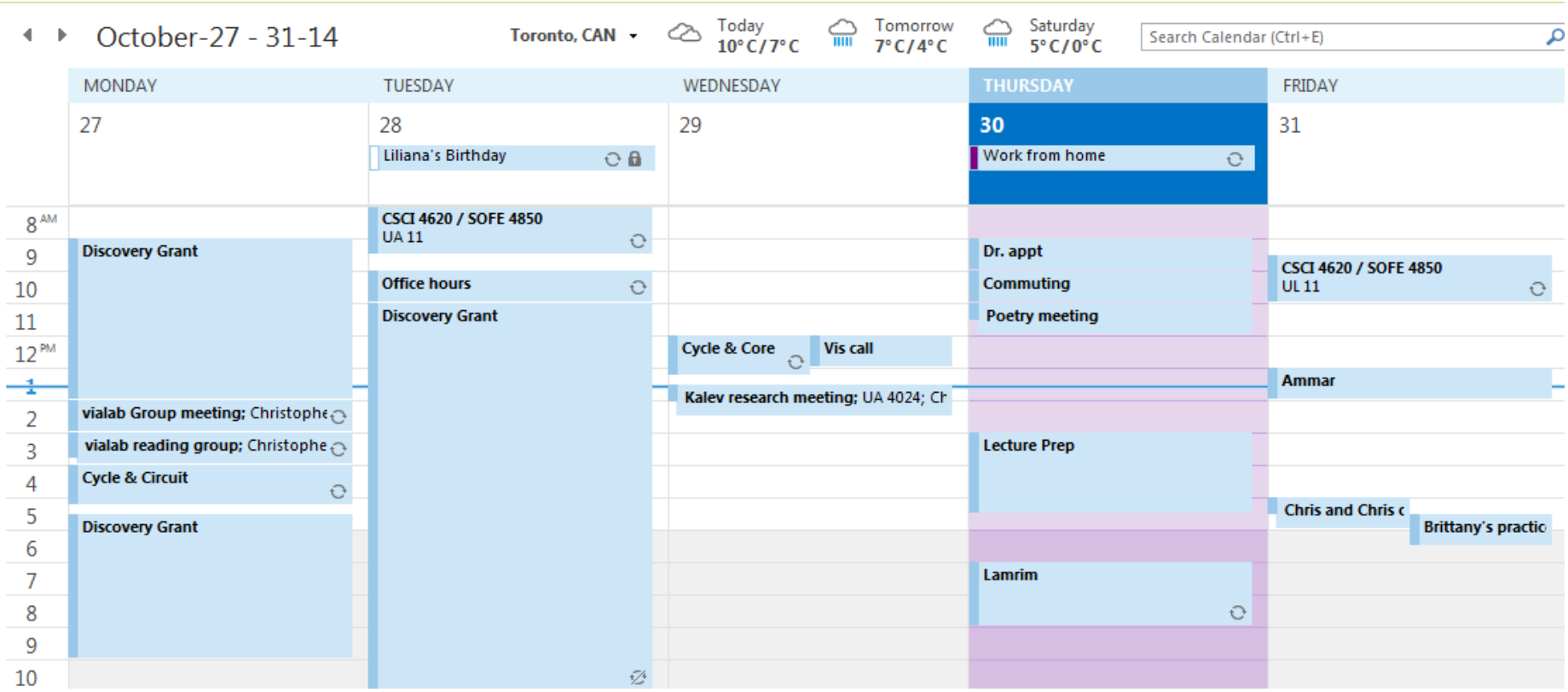
Disk Utilization

WinDirStat

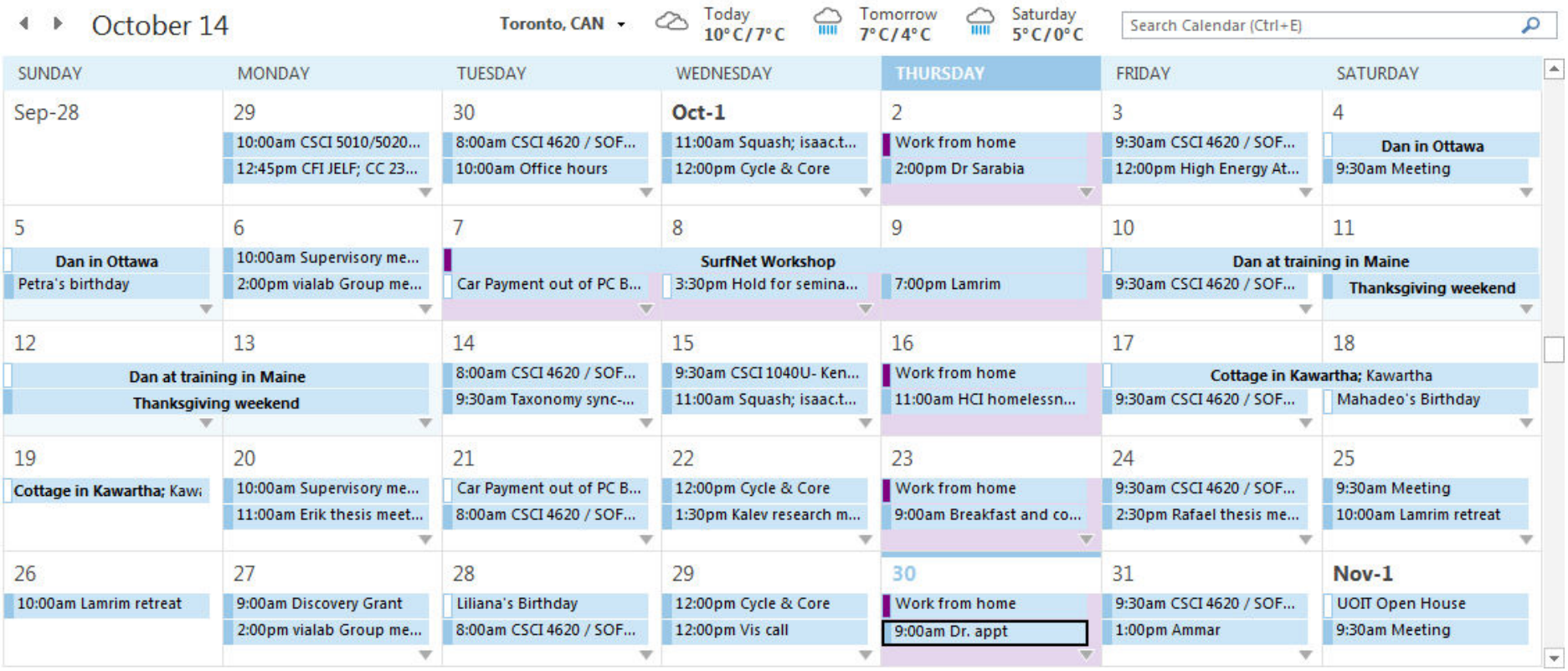


Disk Space Fan

What does my week look like?



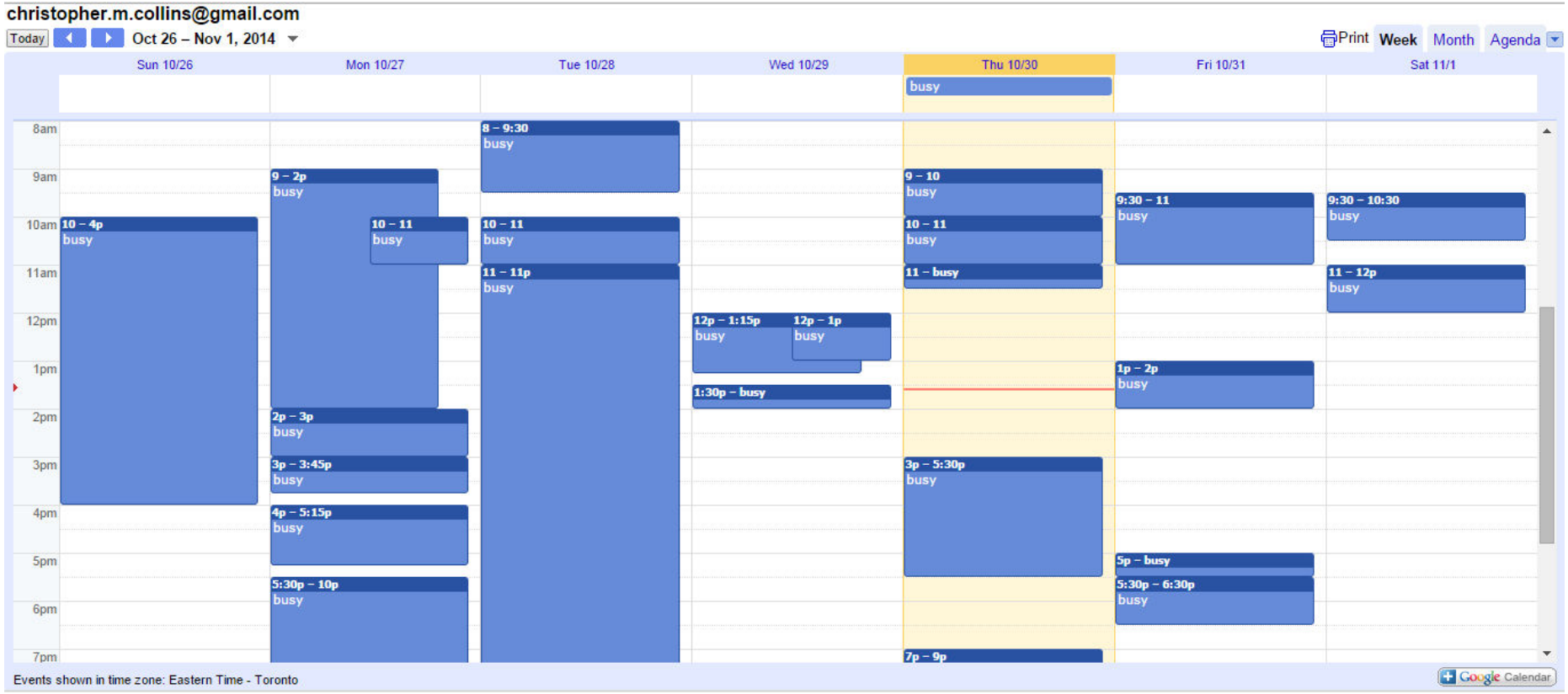
When does my day start?



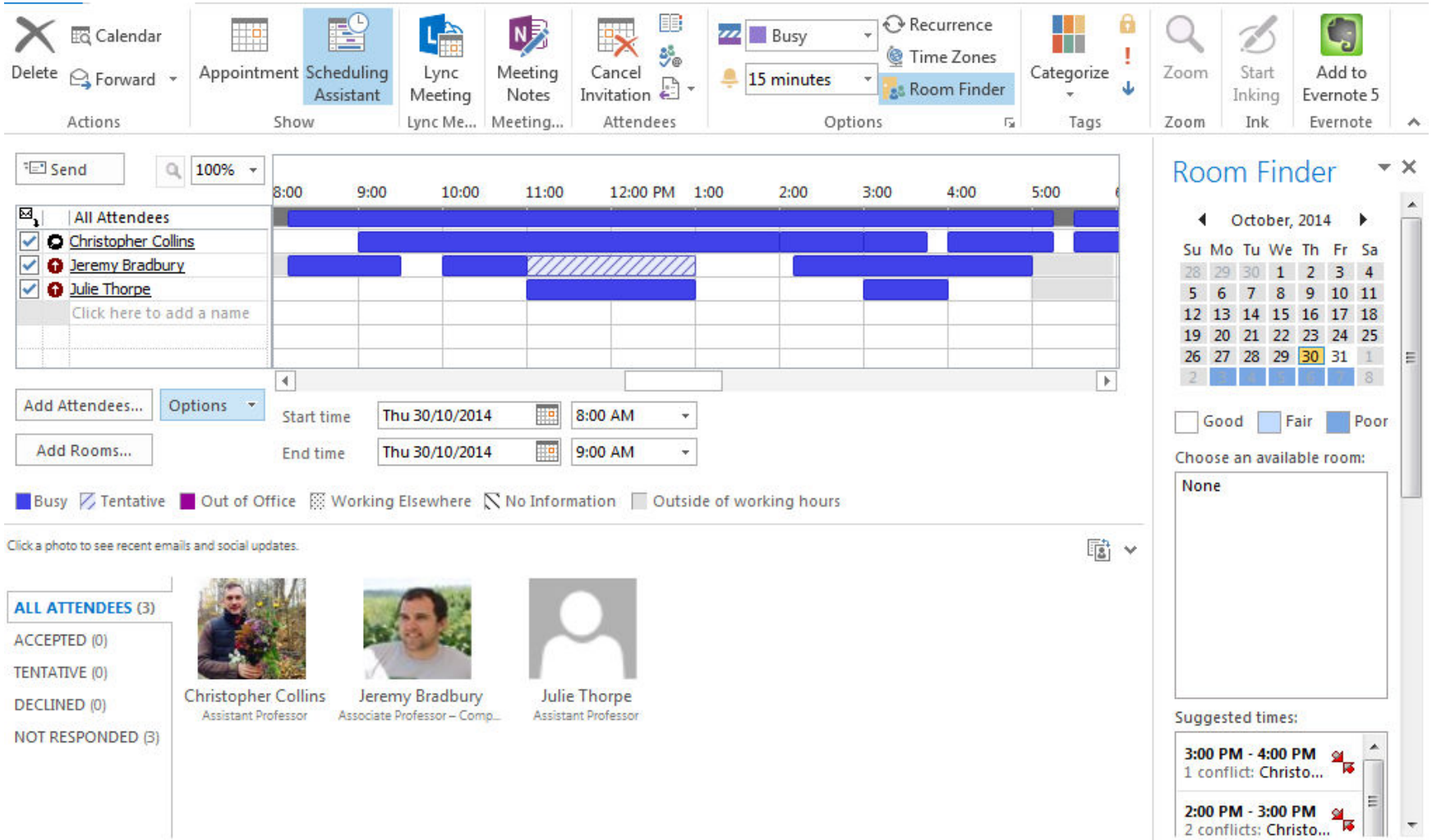
What do I have to do?

	Thanksgiving Day	Canada	Mon 13/10/2014 12:00 AM	Tue 14/10/2014 12:00 AM
	Taxonomy sync-up	your place	Tue 14/10/2014 9:30 AM	Tue 14/10/2014 10:30 AM
	Adam visits		Tue 14/10/2014 1:00 PM	Tue 14/10/2014 3:00 PM
	Commuting		Tue 14/10/2014 3:30 PM	Tue 14/10/2014 6:00 PM
	King Lear		Tue 14/10/2014 6:00 PM	Tue 14/10/2014 11:30 PM
	CSCI 1040U- Ken Pu	UA4170	Wed 15/10/2014 9:30 AM	Wed 15/10/2014 10:30 AM
	Squash		Wed 15/10/2014 11:00 AM	Wed 15/10/2014 12:00 PM
	Simon Reynolds, Studiocode	UA 4170	Wed 15/10/2014 3:30 PM	Wed 15/10/2014 4:30 PM
	HCI homelessness group		Thu 16/10/2014 11:00 AM	Thu 16/10/2014 11:30 AM
	Cottage in Kawartha	Kawartha	Fri 17/10/2014 12:00 AM	Mon 20/10/2014 12:00 AM
	Erik thesis meeting	Lab	Mon 20/10/2014 11:00 AM	Mon 20/10/2014 12:00 PM
	Campus Planning Consultation	G wing dining r...	Mon 20/10/2014 3:00 PM	Mon 20/10/2014 6:00 PM
	Midterm grading	vialab	Tue 21/10/2014 9:30 AM	Tue 21/10/2014 12:00 PM
	Poetry meeting		Tue 21/10/2014 10:30 AM	Tue 21/10/2014 11:00 AM
	Madama Butterfly		Tue 21/10/2014 7:30 PM	Tue 21/10/2014 10:00 PM
	Kalev research meeting	UA 4024	Wed 22/10/2014 1:30 PM	Wed 22/10/2014 2:00 PM

When is Chris free?



When to schedule a meeting?



Practical HCI

IMPLEMENTATION CONSIDERATIONS

Overview

- The role of windowing systems in user interface implementation
  - Including architectures for windowing systems
- Discussion of toolkits for implementing user interfaces
- An overview of the event handling system used in Java's Swing toolkit

Windowing Systems

- It is important to discuss user interface implementation in the context of windowing systems
  - These are systems that manage different applications at the same time
  - These are systems that provide independence between the hardware and the applications – remove the need for applications to worry about the hardware devices a user may choose to use

Windowing Systems

- Windowing systems provide independence by using an imaging model and abstract terminals
  - Each application that runs in a windowing system outputs commands to an abstract terminal
    - can also do some input as well
  - The abstract terminal uses the imaging model of the windowing system
    - This imaging model can be translated to different devices by writing device drivers (all applications can use the same driver – this is much better than application-specific drivers!)

ASIDE:

- What are some examples of imaging models?

ASIDE:

- What are some examples of imaging models?
  - Pixels
  - PostScript
  - SVG

Windowing Systems

- Windowing systems manage the applications by coordinating the sharing of the imaging models
  - Each abstract terminal will use the same imaging models simultaneously

Windowing Systems Architectures

- There are different windowing systems and there are different ways to architect such a system:
  - 1. Management of processes is replicated for each application
    - Direct write to hardware
    - PROBLEM: How do different applications share hardware?
    - Example: problems running DirectX and OpenGL applications at the same time

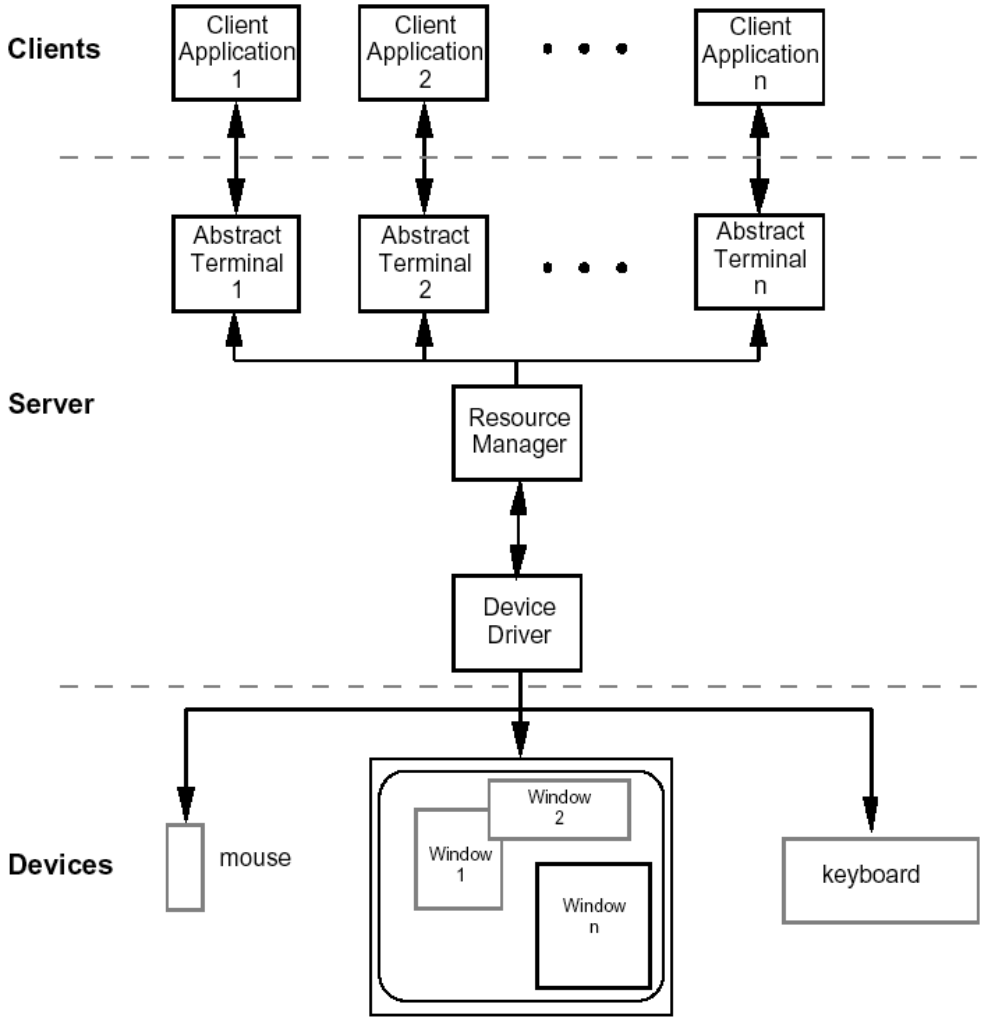
Windowing Systems Architectures

- There are different windowing systems and there are different ways to architect such a system:
  - 2. Management is centralized in the OS kernel
    - This approach resolves the issue of sharing hardware across applications
    - Performance improvements
    - PROBLEM: Applications are now OS-specific

Windowing Systems Architectures

- There are different windowing systems and there are different ways to architect such a system:
  - 3. Management is handled by an application that is OS-independent (client-server)
    - Applications are now portable
    - Can be used with different operating systems over a network
    - X11 / RDP

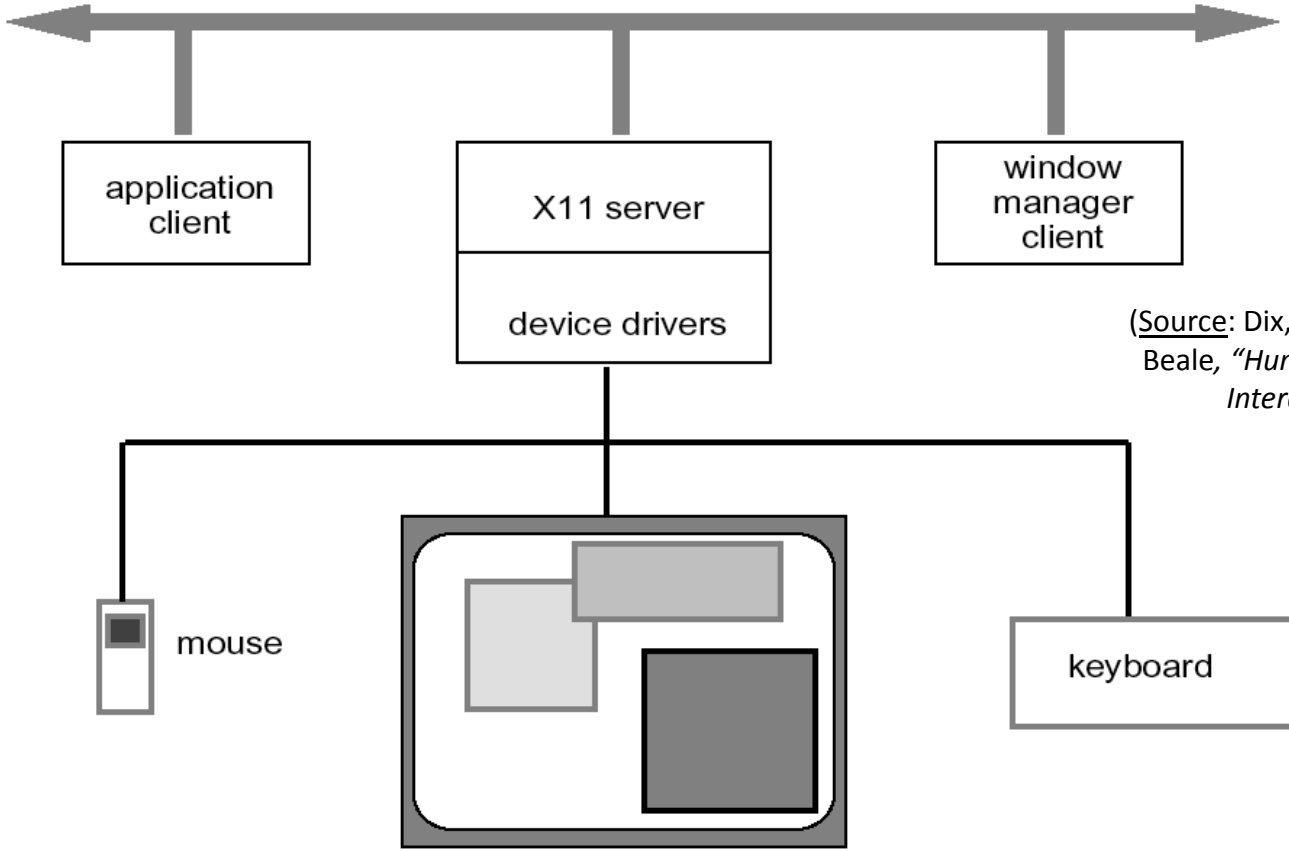
Client-Server Architecture



(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*

Client-Server Architecture

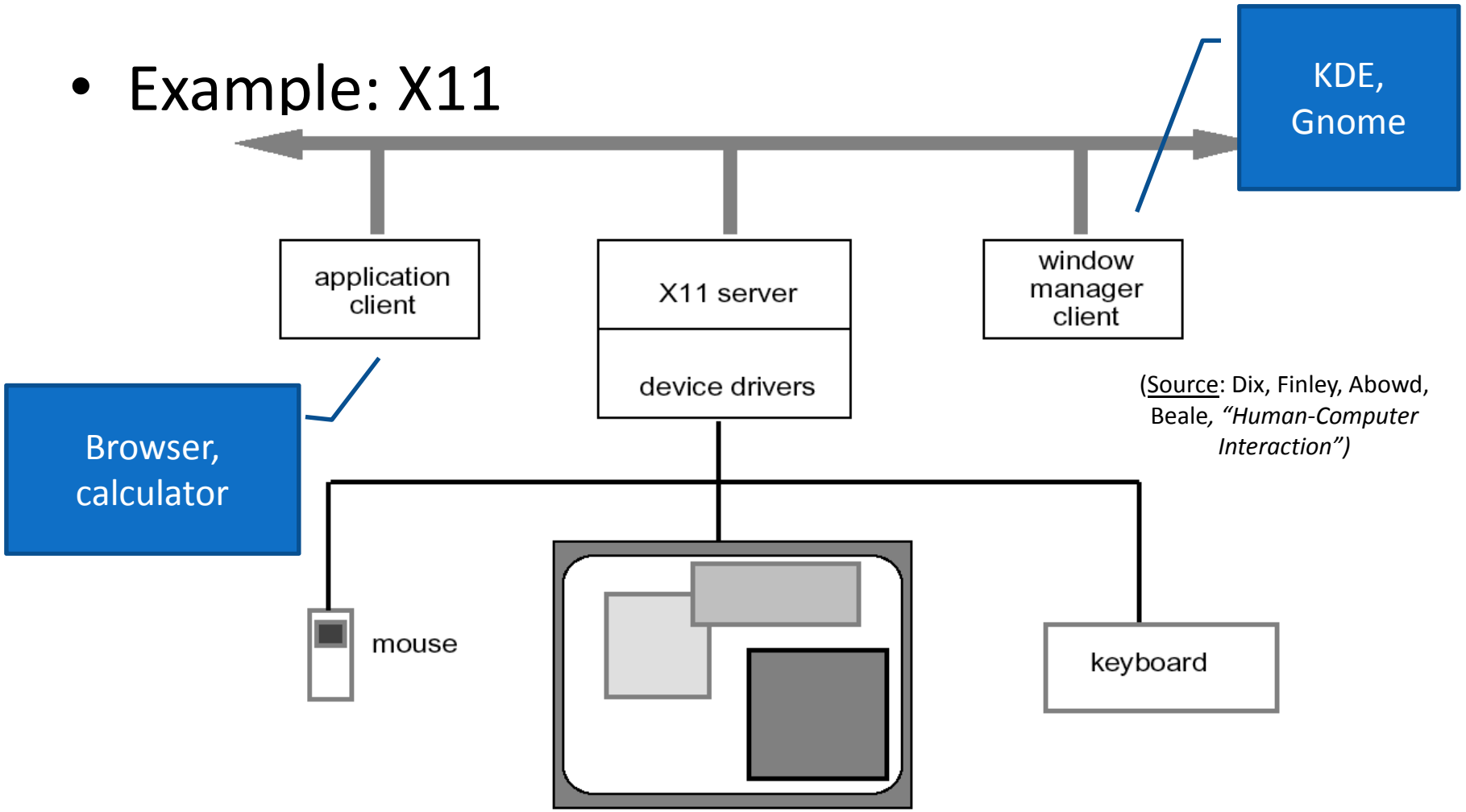
- Example: X11



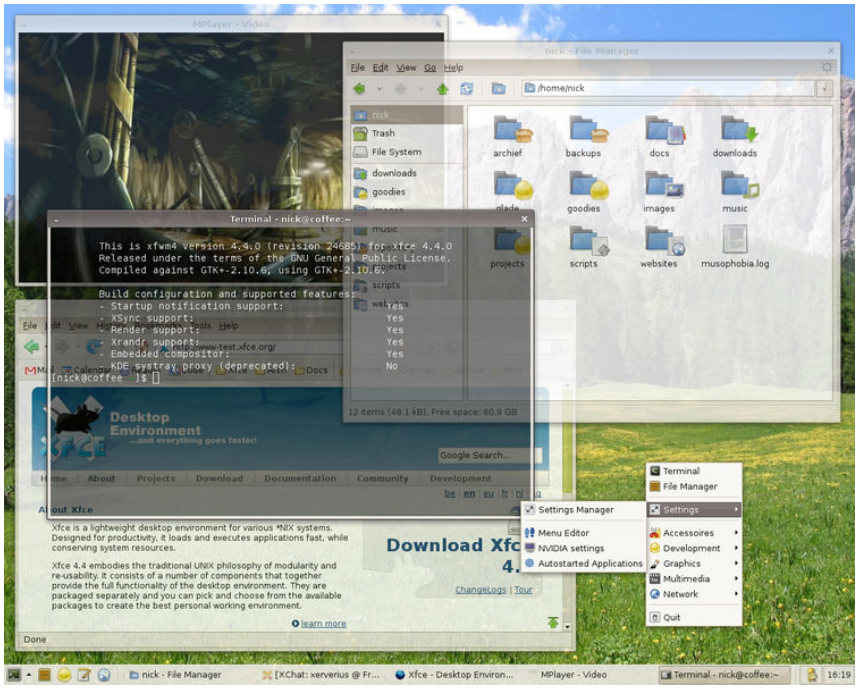
(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*

Client-Server Architecture

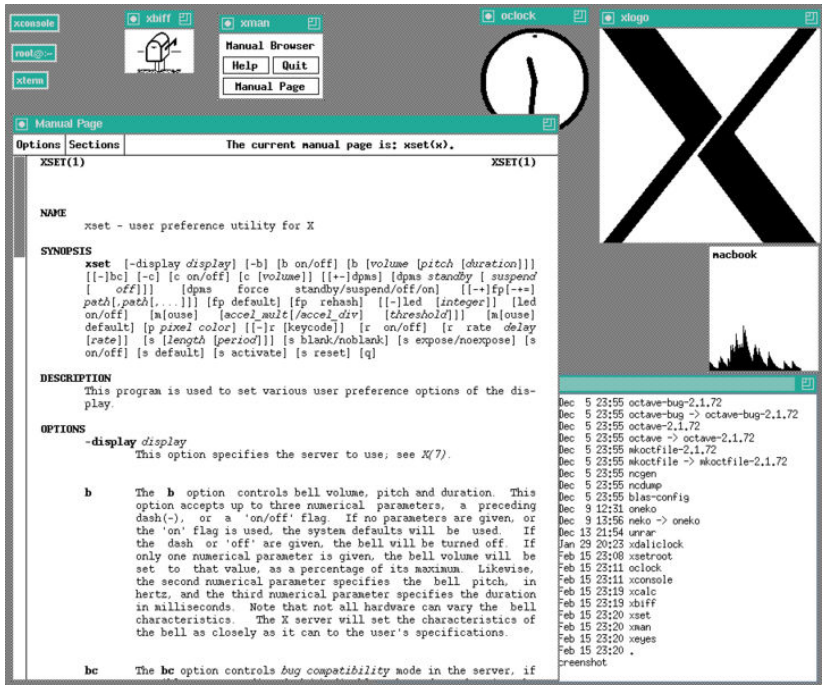
- Example: X11



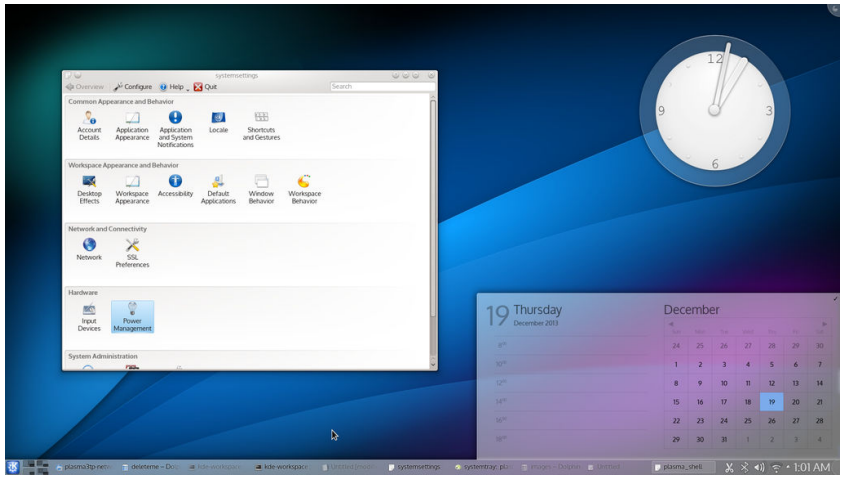
(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*



xfce



twm



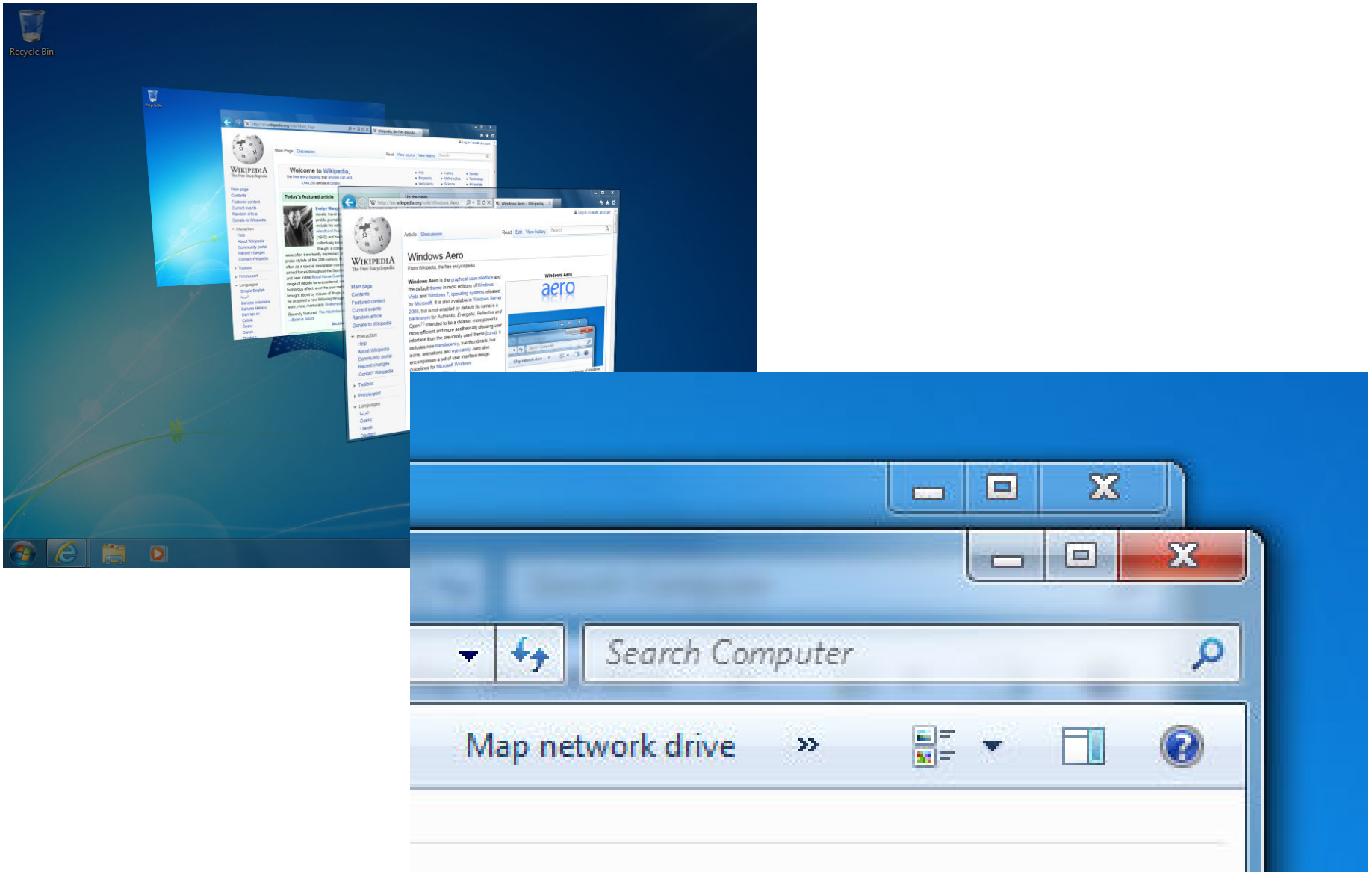
KDE Plasma

Windows “Stacking Manager”

- Pre-Vista
- All windows write to the same buffer in defined regions – write directly to video memory
- Rendering order mattered to handle overlaps
- Background applications are requested to re-render when maximized through WM_PAINT directive
- Not double buffered by default

Windows “Desktop Window Manager”

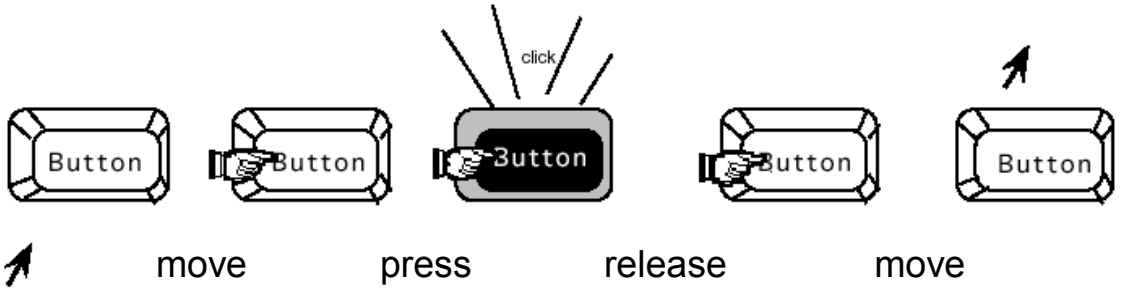
- All applications render to off screen buffers
- The desktop is a full screen Direct3D surface
- Windows are comprised of two mutually inverted triangles onto which window contents are texture-mapped
- Compositing uses hardware shaders
- Enables live previews, blur effects, transparency on screen decorations (the window “chrome”)
- Disabled for full screen games and media applications


Recycle Bin
Windows Aero
Search Computer
Map network drive

HANDLING INPUTS

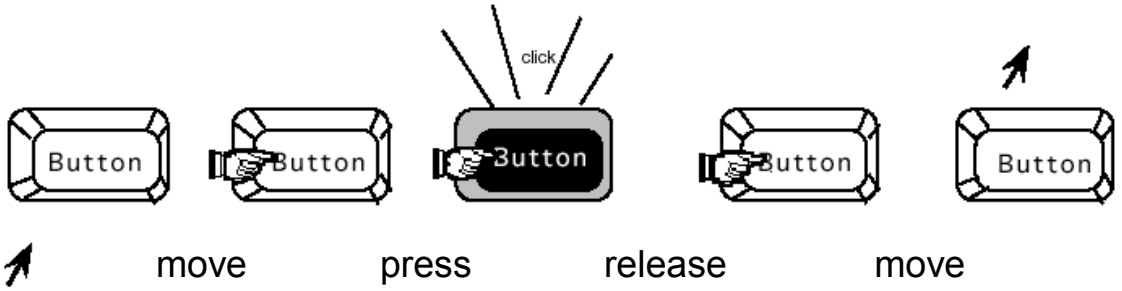
How do we click on a button?

(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*



How do we click on a button?

(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*



- As a programmer we don't worry about animating the button press when the user clicks the mouse, etc.
- This connection between input and output is typically handled for us – How is this done?

Toolkits

- A toolkit sits between the application and the windowing system
  - Why? Convenience!
- A toolkit provides a set of widgets that the programmer can use in his/her application
- A toolkit also provides consistency in the look-and-feel of the interface
- Separation of appearance and behavior

Toolkits

- Toolkits can exist for a specific programming language:
- Example Java Toolkits:
  - AWT – Abstract Windows Toolkit
    - The original Java toolkit
  - Swing
    - A newer toolkit
  - SWT – Standard Widget Toolkit
    - An Eclipse toolkit

Toolkits

- Toolkits can also be platform-specific:
  - MacOS X: Cocoa / Cocoa touch
  - Windows: Windows Presentation Foundation (WPF)
  - Unix: Motif
- Or created for specialized I/O:
  - Simple Multi-touch Toolkit
- Toolkits can be integrated with developer tools (e.g. XAML editor and WPF designer in Visual Studio)

WPF

- resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware
- Extensible Application Markup Language (XAML)
- controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography

WPF

- Scalable vector graphics
- Separation of appearance and behaviour
- Composable, e.g. put images on buttons:


```
<Button>
        <StackPanel Orientation="Horizontal">
                <Image Source="speaker.png" Stretch="Uniform"/>
                <TextBlock Text="Play Sound" />
        </StackPanel>
</Button>
```

A Simple WPF Button


```
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    Title="Window with Button"
    Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button">Click Me!</Button>

</Window>
```

http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx

```xaml
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="SDKSample.AWindow"
    Title="Window with Button"
    Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button" Click="button_Click">Click Me!</Button>

</Window>
```

XAML

C#

```csharp
using System.Windows; // Window, RoutedEventArgs, MessageBox

namespace SDKSample
{
    public partial class AWindow : Window
    {
        public AWindow()
        {
            // InitializeComponent call is required to merge the UI
            // that is defined in markup with this class, including
            // setting properties and registering event handlers
            InitializeComponent();
        }

        void button_Click(object sender, RoutedEventArgs e)
        {
            // Show message box when button is clicked
            MessageBox.Show("Hello, Windows Presentation Foundation!");
        }
    }
}
```

http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx

```xaml
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="SDKSample.AWindow"
    Title="Window with Button"
    Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button" Click="button_Click">Click Me!</Button>

</Window>
```

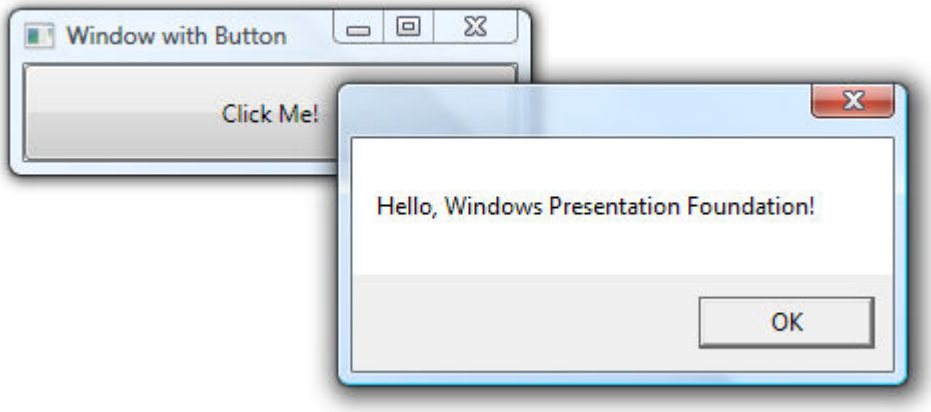
```xaml.cs
using System.Windows; // Window, RoutedEventArgs, MessageBox

namespace SDKSample
{
    public partial class AWindow : Window
    {
        public AWindow()
        {
            // InitializeComponent call is required to merge the UI
            // that is defined in markup with this class, including
            // setting properties and registering event handlers
            InitializeComponent();
        }

        void button_Click(object sender, RoutedEventArgs e)
        {
            // Show message box when button is clicked
            MessageBox.Show("Hello, Windows Presentation Foundation!");
        }
    }
}
```

http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx

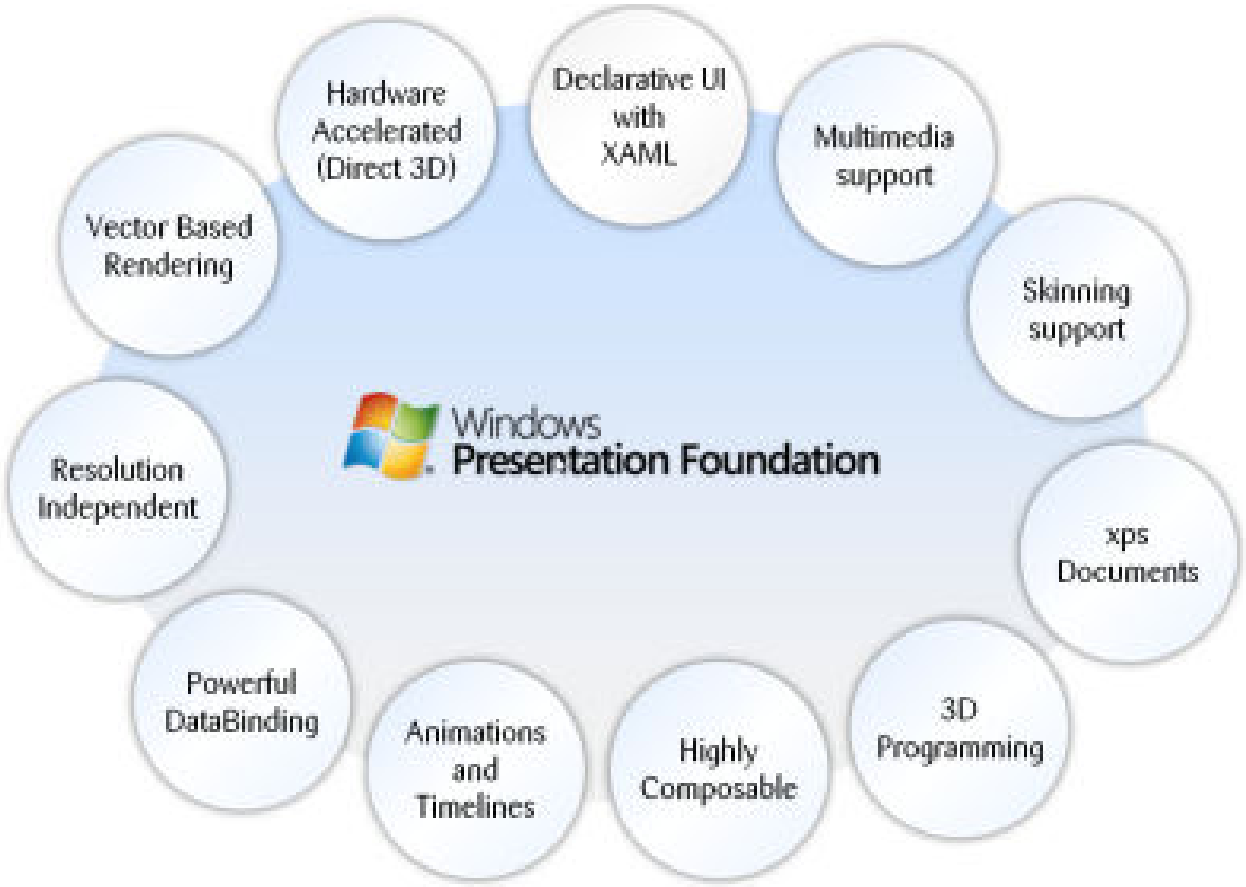
```xaml
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="SDKSample.AWindow"
    Title="Window with Button"
    Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button" Click="button_Click">Click Me!</Button>

</Window>
```

```
using System.Windows; // Window, RoutedEventArgs, MessageBox

namespace SDKSample
{
    public partial class AWindow : Window
    {
        public AWindow()
        {
            // InitializeComponent call is required to merge the UI
            // that is defined in markup with this class, including
            // setting properties and registering event handlers
            InitializeComponent();
```

```
tton_Click(object sender, RoutedEventArgs e)

Show message box when button is clicked
sageBox.Show("Hello, Windows Presentation Foundation!");
```



http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx



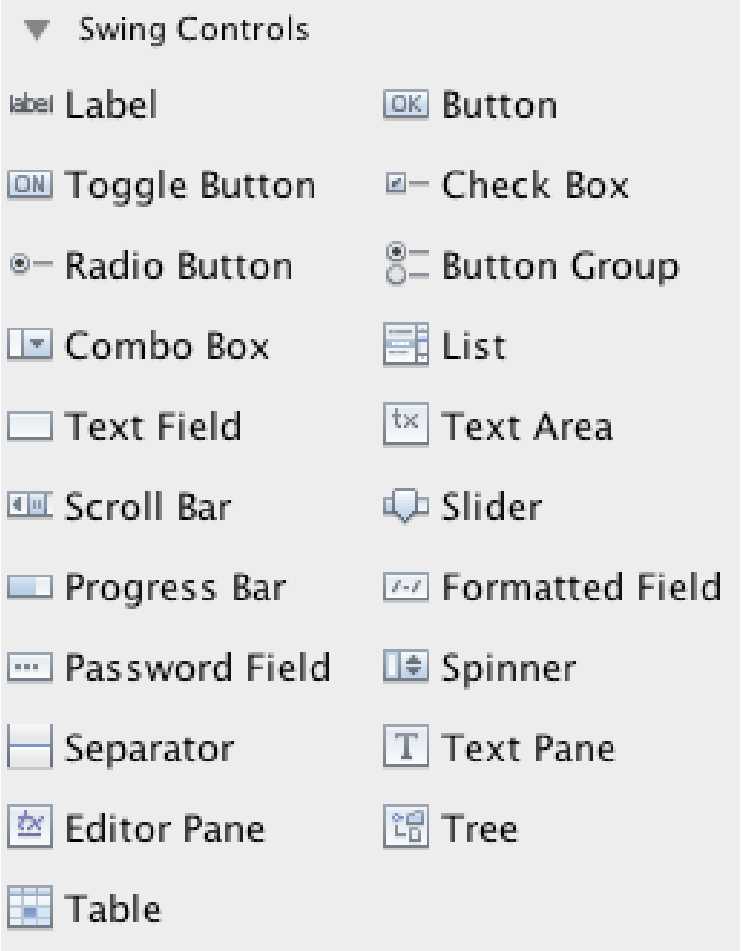
http://www.wpftutorial.net/

Implementation Considerations

EVENT HANDLING IN JAVA SWING

Swing

- Java Swing is a toolkit for interface implementation
- It provides:
  - A collection of widgets
  - A set of layout algorithms
  - A unified event handling system for interaction

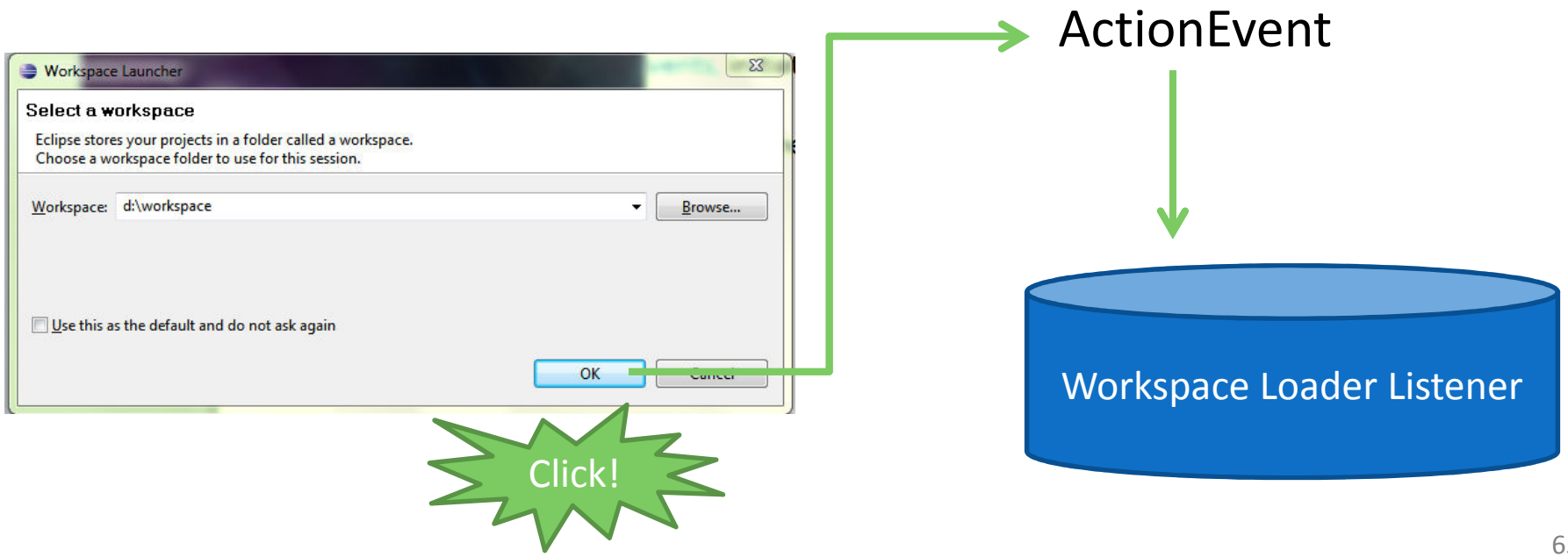


Event Handling

- Event handling is one of the most important aspects of user interface design
  - We use event handling in combination with event listeners
- In addition to ensuring that we have good screen design, it is also important for navigation that something happens when we click on a GUI component!
- We will now look at three approaches to event handling in Java GUI applications

Event Handling

- Interface elements generate events, initiated by user actions.
- Listeners receive events and respond appropriately.



Centralized Approach

- One approach to event handling is to use one method to handle many different event sources
- This approach requires the following:
  - The class implements the ActionListener interface
  - We implement the actionPerformed method from the ActionListener interface
  - We set the current class to be the action listener for all of the components (event sources) in the interface

```
public class centralizedEventHandling implements ActionListener, ChangeListener {
    JButton button1 = new JButton("a button");
    JButton button2 = new JButton("another button");
    JSlider slider1 = new JSlider();

    public void main(String[] args){
        button1.addActionListener(this);
        slider1.addChangeListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button1) {
            // carry out button1 action
        }
        if (e.getSource() == button1) {
            // carry out button2 action
        }
    }

    @Override
    public void stateChanged(ChangeEvent e) {
        if (e.getSource() == slider1) {
            // carry out slider1 changed action
        }
    }
}
```

Inner Classes

- An alternative approach to event handling is to use inner classes to handle the events
  - An inner class for each event source
  - This approach requires that we set the action listener for a given event source to the appropriate inner class

Example:
We could set the okButton to use the inner class okButtonActionListener

```
import java.awt.event.ActionEvent;

public class innerClassEventHandling {
    JButton button1 = new JButton("a button");
    ButtonActionListener buttonActionListener = new ButtonActionListener();

    public void main(String[] args){
        button1.addActionListener(buttonActionListener);
    }

    class ButtonActionListener implements ActionListener {
        int clickCount = 0;
        boolean increase = true;

        @Override
        public void actionPerformed(ActionEvent e) {
            if (increase) clickCount++; else clickCount--;
            if (e.getSource() == button1) {
                // carry out button1 action
            }
        }
    };

    private void changeActionHandling() {
        buttonActionListener.increase = !buttonActionListener.increase;
    }
}
```

Anonymous Inner Classes

- Another alternative approach to event handling is to use anonymous inner classes to handle the events
- This approach works well when we are only interested in having a single instance of a given inner class

```
import java.awt.event.ActionEvent;

public class anonymousClassEventHandling {
    JButton button1 = new JButton("a button");

    public void main(String[] args){
        button1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // carry out button1 action
            }
        });
    }
}
```

```
import java.awt.event.ActionEvent;

public class anonymousClassEventHandling {
    JButton button1 = new JButton("a button");

    public void main(String[] args){
        button1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // carry out button1 action
            }
        });
    }
}
```

Event Handling

- We have considered the following three approaches to event handling in Java Swing user interfaces:
  - Centralized approach
  - Inner classes
  - Anonymous inner classes
- Advice: where possible, use anonymous inner classes

Event Handling

- We have considered the following three approaches to event handling in Java Swing user interfaces:
  - Centralized approach
  - Inner classes
  - Anonymous inner classes
- Advice: where possible, use anonymous inner classes
  - Why?

Event Handling

- We have considered the following three approaches to event handling in Java Swing user interfaces:
  - Centralized approach
  - Inner classes
  - Anonymous inner classes
- Advice: where possible, use anonymous inner classes
  - Code is more maintainable, extendable, readable, etc.

Java Swing Events

- Java Swing widgets such as JButton produce an ActionEvent
  - ActionEvents are neutral to the type of interaction that caused the event – click a button or use tab and space bar to activate it: both generate the same ActionEvent
- Interaction specific events can also be captured: MouseEvent, KeyEvent, TouchEvent

Interface vs. Adapter

- MouseListener Interface requires implementation of all mouse event methods:
  - mouseClicked()
  - mouseEntered()
  - mousePressed()
  - mouseReleased()
- MouseAdapter has empty implementations of these, so just override what you want to use!

Processing

- In *Processing* the Sketch itself acts as an adapter class. It has (hidden) empty implementations of many event-processing methods.
- In your sketch you can implement:
  - mouseClicked()
  - mouseDragged()
  - keyPressed()
  - touch() [when importing SMT library]
  - …
- Or you can ignore them! It just works!

Swing & MVC

- Swing provides a degree of separation between the model, view and controller through the use of models (e.g., TableModel)

Summary

- Today we introduced:
  - Window Managing Systems
  - Toolkits for Interface Building
  - Event Handling, with emphasis on Java Swing

Suggested Reading

- The Java Tutorials: Creating a GUI with JFC/Swing, http://java.sun.com/docs/books/tutorial/uiswing/index.html
- Getting Started with WPF *http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx*

Your Action Items

- Finish project part 3a
- Read through group project part 3b
- READINGS:
  - Snyder on prototyping (posted on Blackboard)

Ongoing Course Evaluation

- Please complete the Lecture 14 feedback form if you have comments!