CSE 40416
System Interface Design

Prof. Aaron Striegel
Department of Computer Science & Engineering
University of Notre Dame

Lecture 8 – September 11, 2009

Today’s Lecture

- Clip – Project Natal
- Interface Design
  - Iterative / Task Design
- Primer – WPF
  - Two weeks – basics
  - Start on Surface SDK on Monday in class

Reminders
- Blog Post (Week)
- Project 1

Next Week

- Industry Week
  - Get your resumes ready
- Guest Lecture – Wednesday
  - ND Alums @ Microsoft – Surface
- Lab Session - Friday
  - 355S – Live Surface Demos / Coding
Project Natal

http://www.youtube.com/watch?v=oAC19R9z37U

Small Group Exercise

☐ What questions might you have about the technology?

Split into groups of 2-4 students

Refresher - Iterative Design

Diagram with steps: Design, Evaluate, Implement, Design again.
Think Spiral Design

- **Design**
- **Evaluate**
- **Implement**

Start with low fidelity (prototype) Work outwards w/feedback

---

Think Spiral Design

- **Design**
- **Evaluate**
- **Implement**

1. Task Analysis
2. Design Sketches
3. Paper Prototype
4. User Eval
5. Computer Proto

We’ll do in-class evaluations for the project

---

Case Study – Olympic Msg System

- Cheap Prototypes
  - Scenarios
  - User guides
  - Simulation
  - Prototyping tools
- Iterative design
  - 200 for user guide
- User
  - Non-native speakers vs. alphabet
User Classes

- Applications – multiple users
- Example: Olympic Messaging System
  - Athletes
  - Friends & family
  - Telephone operators
  - System admins

Task Analysis

- Identify individual tasks
- Task -> goal
  - What vs. how
  - Don't focus on solution, focus on task
- Hierarchical decomposition
  - Goal: Shoppers pay for own groceries
    - Enter groceries into register
    - Bag groceries
    - Pay

Task Analysis (2)

- What needs to be done?
  - Goal
- What must be done first to make it possible?
  - Preconditions
    - Dependencies
    - Information to user
- What steps are involved?
  - Subtasks
  - May use recursion
Other Task Questions

- How is the task performed?
  - Front of supermarket, standing up
- How often is the task performed?
  - 1x or 2x per week
- What are its time or resource constraints?
  - A minute or two
- How is the task learned?
  - By trying it
  - By watching others
  - By being shown how

Other Task Questions

- What can go wrong?
  - Barcode – missing or smudged
  - Shopper wants to buy alcohol or cigarettes
- Who else is involved?

Pitfalls of Task Analysis

- Staying in the box
  - Duplicating a bad existing procedure
- There is no box
  - Ignoring good aspects of existing procedure
- Focus
  - Why vs. what
**Contextual Inquiry**

- Observe users doing real work in the real work environment
- Be concrete
- Establish master / apprentice
  - User shows how and discusses
  - Interviewer watches / ask questions
- Challenge assumptions
- Probe surprises

**Exercise – Alumni Association**

- Project we will do later in the semester
- Premise:
  - Visitors over at the Eck Center
- Discuss:
  - Tasks that the user wants

Split into groups of 2-4 students

**Windows Presentation Foundation**

- Highly recommend you pick up a book
  - WPF Unleashed
- WPF API Aspects
  - Blends DirectX (HW Accel) + Windows Forms
  - Web-like layout
  - Rich text / drawing models
  - Animation = first-class concept
  - Intrinsic audio / video support
  - Styles / templates
  - Declarative user interface (XAML)
  - Resolution independence
General Architecture

PresentationFramework.dll
PresentationCore.dll
WindowsBase.dll
Milcore.dll
Media Integration Layer
Direct3D
User32

Class Hierarchy

DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Shape
Control
Panel
ItemsControl
ContentControl

XAML (Zammel)

- eXtensible Application Markup Language
- Code
  - By hand
  - Tool – Expression Studio
- Our code
  - XAML component – controls
  - C# component – logic / glue
### XAML Basics

- Every element -> .NET class
  - Core .NET framework
  - Our own classes
  - `<Button>` -> Makes a button
- Nesting, nesting, nesting
- Express containment
- Properties
  - Attributes
    - Height = "300"
  - Special syntax

### Example

```xml
<Window x:Class="WindowsApplication1.Window1"
    xmlns="http://...
    xmlns:x="http://...
    Title = "Window1" Height="300" Width="300"

    <Grid>
    </Grid>
</Window>
```

### Questions?

- Week 3 – Blog
- Project 1