Interaction Design

DECO1200
Outline

Introduction to Conceptual Models
  What are conceptual models? Why should you care?

Conceptual Models Based on Activities
  Supporting common interactive activities.

Conceptual Models Based on Objects
  Designing interactive systems to mimic real objects.

Conceptual Models Based on Metaphors
  Designing interactive systems that are like other things.

Interaction Paradigms
  Broad classes of conceptual models.
Getting Started

Where does interaction design start?

The User Interface?

The System Architecture?

The Computer Program?

None of the Above

Interaction design starts with the user...
Failing to ask the user what they want can lead to the whole crops being lost...
What’s the Problem?

Resist the temptation to use the latest cool technology or spend hours creating beautiful graphics before really understanding what the design problem is to be solved.

This doesn’t mean that you can’t use the latest cool technology or spend hours creating beautiful graphics later on, just make sure that you understand the design problem first.

Will your design support the user? How?

What is the problem you are try to solve?
What are the important issues for the user?
Checking Assumptions and Claims

What are the problems with a product?
  Are there any problems with an existing product?
  If so, what do the users think are the problems?
  What do users think cause the problems?

How will your proposed design work?
  How will it support people in their activities?
  How does it address the identified problems?

Why will your proposed design be useful?
  How will users integrate it into their lives?
Example: WAP

WAP (Wireless Application Protocol)

WAP enabled people to connect to Internet services through their mobile phone with very small displays.

Assumptions

People want to stay informed wherever they are.
People want to interact while on the move.
People want to make decisions on the move.
People will be happy using a very small screen.

How reasonable are these assumptions?
Conceptual Models

A description of a proposed system as a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by its users in the manner intended.
Conceptual Models Based on Activities

Instructing
User issues instructions to the system, e.g. command line interfaces, computer-aided design systems

Conversing
User converses with the system, e.g. phone-based menu systems, interactive agents

Manipulating and navigating
User manipulates aspects of the system, e.g. desktop, virtual environments, videogames

Exploring and browsing
User explores the system’s structure, e.g. books, CDs, help systems, the web
Activities

What activities do the following products support?

iTunes Music Store
Photoshop
AIM/MSN Messenger
Conceptual Models Based on Objects

Conceptual models can be based on objects that users are familiar with in the context. Conceptual models based on objects provide users with clues about what a product does and does not support.

Often the conceptual model provides a good starting point for exploring a new product. Typically, a new product will have additional features when compared to the object on which they are based.

Sometimes the conceptual model needs to simulate an existing object or system. Systems that are designed for training need to capture all of the important aspects e.g. flight simulators.
Example: VisiCalc

VisiCalc was a revolutionary application for the Apple and IBM PCs (the first “killer app”)

Released in 1979, VisiCalc was one of the first apps to make computers useful for “normal people”

VisiCalc was based on a paper ledger sheet, a familiar objects for people in finance

Dan Bricklin knew how accountants etc. worked and also knew what the problems with other software

VisiCalc could do lots of things that could not be done with a ledger sheet

e.g. automatic calculations of cell values etc.
Example: Xerox Star

In 1981, Xerox introduced the Star system that revolutionised personal computing. The Star system wasn’t commercially successful but Apple and Microsoft “borrowed” extensively from it.

Much of the Star user interface was based on familiar objects in an office:
- e.g. documents as pieces of papers,
- directories as folders,
- in trays for e-mail,
- trash cans, etc.
XEROX 8010 Star Information System

Star provides integrated text and graphics. A variety of type sizes and styles may be used.

![Diagram showing various shapes and icons]

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peas</td>
<td>$0.39</td>
</tr>
<tr>
<td>Beans</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

NAME       SIZE             VERSION OF
0 Temporary title page  1 Page  10/30/84 13:29
A Copyright and abstract  2 Pages  11/06/84 16:12
B Front matter  4 Pages  10/31/84 22:05
Chapter 1  11 Pages  10/30/84 13:56
Chapter 2  15 Pages  10/31/84 22:49
Chapter 3  21 Pages  11/02/84 15:41
Chapter 4  36 Pages  10/31/84 21:47
Chapter 5  15 Pages  11/02/84 15:45
Chapter 6  7 Pages  10/30/84 18:02
Chapter 7  13 Pages  10/31/84 22:10
References  3 Pages  10/31/84 21:56
Styles  5 Pages  10/22/84 11:42

Thesis

NAME

A Copyright and abstract
B Front matter
Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Chapter 7
References
Styles

SIZE

1 Page
2 Pages
4 Pages
11 Pages
15 Pages
21 Pages
36 Pages
15 Pages
7 Pages
13 Pages
3 Pages
5 Pages

VERSION OF

10/30/84 13:29
11/06/84 16:12
10/31/84 22:05
10/30/84 13:56
10/31/84 22:49
11/02/84 15:41
10/31/84 21:47
11/02/84 15:45
10/30/84 18:02
10/31/84 22:10
10/31/84 21:56
10/22/84 11:42
Conceptual Models Based on Metaphors

(AKA: Interface Metaphors)

Conceptual models based on metaphors to existing objects or systems that are similar

Products designed to be similar in some way to aspects of an existing object or system but that has its own behaviours and properties

Metaphors are meant to provide a hint at the essence of a product or process

Metaphors provide clues about how something works in a general sense that users can use to explore further
Example: Spiders

Automatic systems that explore the World Wide Web are often referred to as “spiders”
Spiders are used by search engines (e.g. Google) to index web pages and record the links between them

These systems autonomously explore the system by following the links between pages
The metaphor is that these systems are like spiders following the threads that make up their web

NOTE: You don’t have to understand the details to get a sense of what spiders do.
Why are Interface Metaphors Popular?

Interface metaphors are popular because they can help explain difficult concepts sufficiently in terms of something that is familiar.

Computer systems are often complex and have no direct analog in the real world.

Interface metaphors can help explain these concepts without lots of technical details.
Obscure Metaphors

Sometimes interface metaphors are very obscure and little more than clever naming.

Obscure names are often found in highly-technical computer systems, hackers love obscure metaphors.

Example: Bluetooth is a wireless technology that allows many devices to communicate.

Bluetooth gets its name from a Viking King, King Harald Blue Tooth, who united Scandinavia allowing many people to talk to each other.

Obviously, obscure metaphors don’t help the user but they can be fun to learn about.
Problems Using Interface Metaphors

Breaking the rules: metaphors often break logical/cultural rules when implemented

Too constraining: using metaphors can make simple tasks complicated, e.g. finding a file

Conflicting with design principles: using a metaphor can conflict with good design

Too limiting for the user: using a metaphor can limit what the user understands about the system

Adopting bad ideas: metaphors can bring along a lot of unwanted “baggage”

Too limiting for the designer: using a metaphor can limit what the designer can imagine
Communicating
Conceptual Models

Design Model
Designer

User Model
User

System
System Image
Interaction Paradigms

Ubiquitous Computing: embedding technology into the environment to “invisibly” enhance our world

Pervasive Computing: seamless integration of technology allowing access information anywhere and at anytime

Wearable Computing: technology integrated into clothes, jewelry, glasses, watches, etc.

Tangible User Interfaces: embedding technology into physical objects to support natural interactions

Augmented Reality: superimposing digital information on the physical world to make it more accessible

Attentive Environments: getting computers to pay attention to people so they can anticipate what users will need
Think about the problem and identify some initial user requirements

Gather data about problems and issues from potential users

Determine requirements and generate some conceptual models

Generate prototypes and whether the conceptual models help users
Summary

Introduction to Conceptual Models
Conceptual models provide consistency

Conceptual Models Based on Activities
Instructing, conversing, manipulating, browsing

Conceptual Models Based on Objects
Models that mimic familiar products.

Conceptual Models Based on Metaphors
Models that draw similarities to familiar products.

Interaction Paradigms
Ubiquitous computing, augmented reality, etc.