Conceptual Design and Mental Models

Human Computer Interaction Research Group
Department of Computer Science

Yay Design!

- So with your scenarios you have a basic idea of what kind of interactive system(s) you want to introduce … inevitably you ask …

Can I start coding yet?

But what would you program anyways?

- When we wrote our scenarios we stayed very neutral in implementation details – that is because we really don't know a lot about how we are going to implement the technology
- Before you start physically designing something, you need to understand what the human and the technology are bringing to the table

But why would we care about the user?

- There is a pervasive myth of the ‘dumb user’ in computing – that users are essentially incompetent knuckleheads who intentionally screw up the system
- In reality, almost always, it is dumb design – the system has been constructed in such a way that it not only doesn’t take the human into account – it actively thumbs its nose at the user
- In some cases it is lack of training – but even in extremely complex systems training should be limited to complex actions – not the basic understanding of what the system is intended to do for the user

People bring a lot with them

- There are lots of things in the real world that people know how to use
- Let's use sport as an example – what do you learn from Cricket?
  - How to bowl
  - How to hit
  - Fielding positions
  - Social norms
  - Penalties
  - …
- That is a lot of stuff – if humans were the 'dumb user' we could hardly do things like play Cricket
- So why can we do things like play Cricket but not use interactive systems?

Why can we play Cricket?

- The reason we can play any sport is because we have a clear view of the concepts (e.g. bat, ball, pitch), the relationships between those concepts (e.g. ball is bowled down the pitch to the bat), and the rules of how they work together (e.g. you can have two batters, one at each end)

What can we draw on from the scenario?

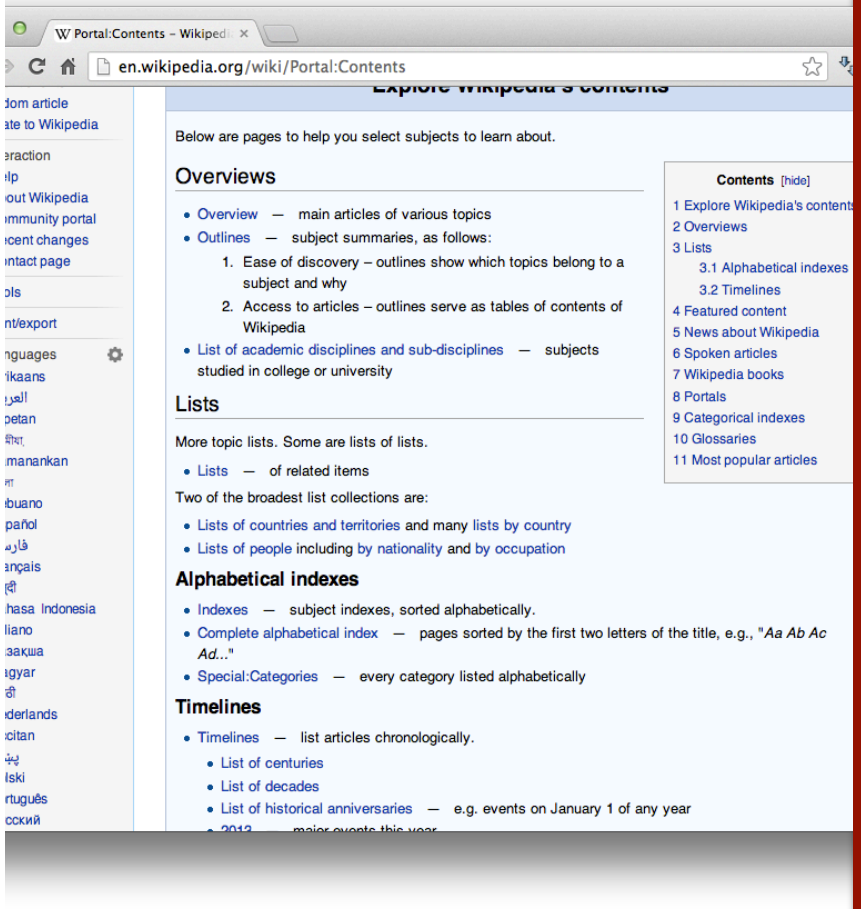
- In our scenario we have a couple of different things that we can use:
  - Personas – tell us specifically who will use the system, so we can explore how they think about the world they are living in
  - Scenario tasks – tell us what the person is going to do in the system
  - Concepts – tell us what types of things the person will need to know

Let’s take an example …

- Wikipedia
  - Why is Wikipedia so popular?
- Part of it is – it is easy to learn, easy to remember, and let’s the person accomplish a goal
  - What do all of these thing relate to?
- How does that help us? Why is Wikipedia all of those things?

Concepts

- Wikipedia is a perfect example of a “web” interactive system
- It has three things that users need to know about
  - Content
  - Pages
  - Links
  - Overviews
  - Lists
- Everything else is just implementation details – at the heart of it there is very little for the user to understand in terms of concepts about Wikipedia



Comparison: Concepts in Facebook

- Facebook (while incredibly popular) is more difficult to learn and use than Wikipedia
- What do users of facebook need to know about?
  - Content
  - Links
  - Pages
- That's the same as Wikipedia – what else?
  - Status messages, Notes, Friends, Groups (at one time Networks), Privacy settings, Events, Ads, Lists, Apps
  - And soooooo much more …

How do we use Concepts?

- Using our scenarios we can identify things that users need to know about
- If that list is really long, or seems at odds with our personas, then we might need to rework our concept load on the user – we might be trying to do too many things
- Some concepts we can borrow from other applications and use them – because they match the *mental model* about how the world works for the user

Mental Models

- Mental models are description of how people think (from cognitive psychology Kenneth Craik, 1943)
- More specifically – it is the understanding that people have about concepts and relationships in the real world that they use to make decisions or react to events
- People try to apply their mental models to different situations – sometimes with varying levels of success
- In terms of Norman's model of action, they are used during the evaluation state to plan the next goal, and in the intention state to choose the appropriate course of action

A well known example …

- If I walk into the house and it is really cold – I want the house to get as warm as possible as fast as possible – how high should I set the thermostat?
  a. Well below the temperature I want
  b. Just below the temperature I want
  c. At the temperature I want
  d. Just above the temperature I want
  e. Well above the temperature I want

Violation of Mental Models

- What happens when a system violates the mental model of the user?
  - The person will not know what to do next (unable to form the next goal)
  - The person will make an error (provoked or unprovoked) – leading to badness

A lesser known example …

- Canadians drive on the left hand side of the car and on the right hand side of the road
- Britons drive on the right hand side of the car and on the left hand side of the road
- What are some of the mental model violations that exist for a Canadian who is driving in the UK?
- Violation of those mental models leads to what is called *negative skill transfer*

So what have we learned …

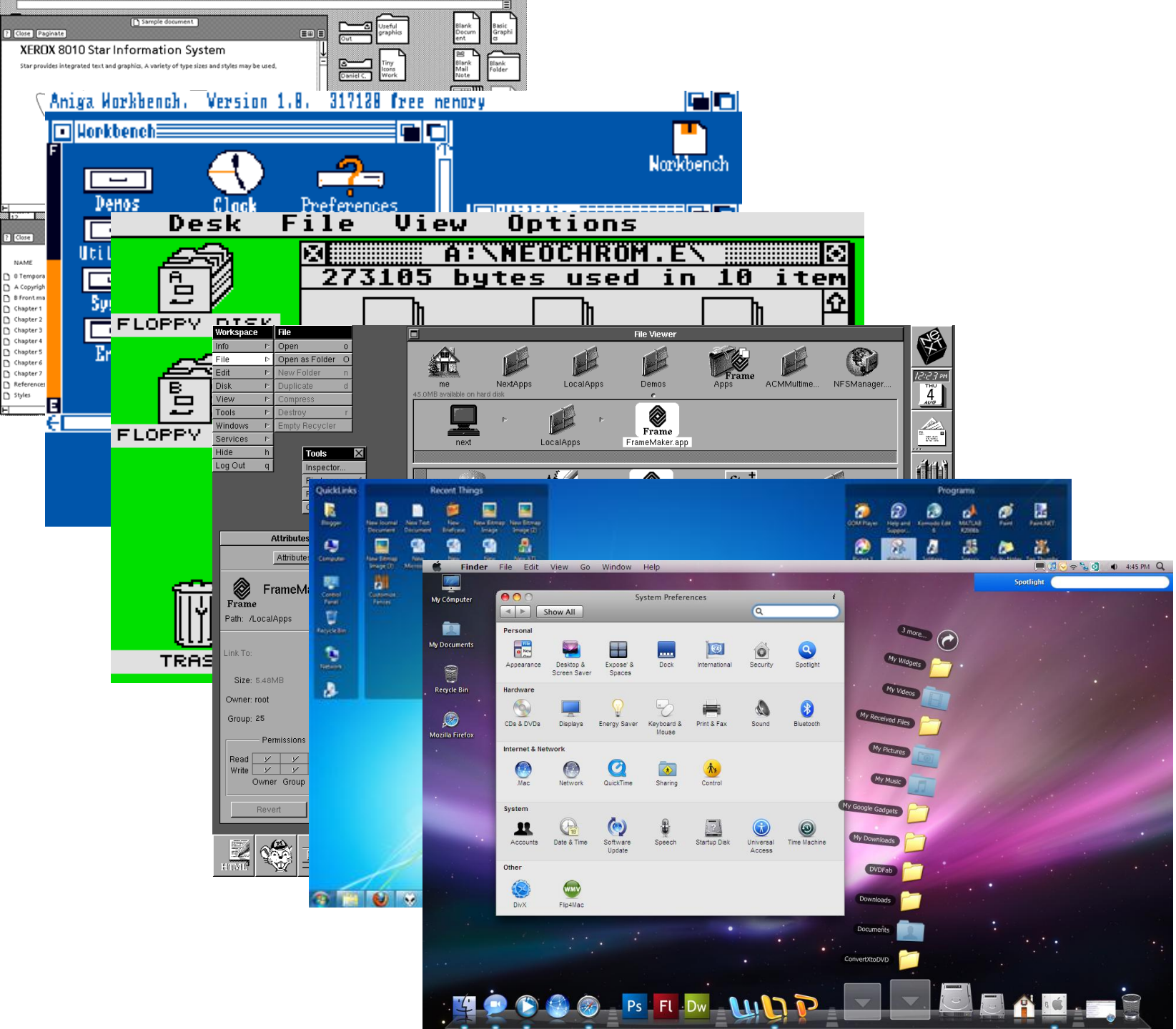
- In order to have a good system that is easy to learn and remember, it would be good if we had:
  - A clear set of concepts that can be applied to an application
  - A clear set of relationships that describe how those concepts relate to one another
  - A mapping from the concepts and relationships we want to use to a mental model of the users who will be using our system
- That sounds really really hard … what can we do to make it easier?

Metaphors

- We have lots of rules for real life – why can't we borrow some of those for design?
- The notion of a *metaphor* in interaction design is the translation of the physical to the digital – where you borrow concepts and rules so that users can apply their knowledge of the physical world

The most famous metaphor?

- There is one particular metaphor that has dominated personal computing for 20 years


XEROX 8010 Star Information System
Amiga Workbench. Version 1.8. 317128 free memory
Workbench
Demos
Clock
Preferences
Desk File View Options
A:\NEOCHROM.E\
273105 bytes used in 10 item
FLOPPY
Workspace
File
Info
Open
Open as Folder
New Folder
Duplicate
Compress
Destroy
Empty Recycler
Edit
Disk
View
Tools
Windows
Services
Hide
Log Out
Inspector...
File Viewer
me
NextApps
LocalApps
Demos
Frame Apps
ACMMultime...
NFSManager...
45.0MB available on hard disk
next
LocalApps
FrameMaker.app
QuickLinks
Recent Things
Programs
Attributes
Path: /LocalApps
Link To:
Size: 5.48MB
Owner: root
Group: 25
Permissions
Read
Write
Owner Group
Revert
Finder File Edit View Go Window Help
4:45 PM
Spotlight
My Computer
My Documents
Recycle Bin
Mozilla Firefox
System Preferences
Show All
Personal
Appearance
Desktop & Screen Saver
Expose' & Spaces
Dock
International
Security
Spotlight
Hardware
CDs & DVDs
Displays
Energy Saver
Keyboard & Mouse
Print & Fax
Sound
Bluetooth
Internet & Network
.Mac
Network
QuickTime
Sharing
Control
System
Accounts
Date & Time
Software Update
Speech
Startup Disk
Universal Access
Time Machine
Other
DivX
Flip4Mac
3 more...
My Widgets
My Videos
My Received Files
My Pictures
My Music
My Google Gadgets
My Downloads
DVDFab
Downloads
Documents
ConvertXtoDVD

Desktop Metaphor

- The Desktop metaphor changed the way the game was played
- Now people could think about their computer the same way they did about their desk
  - Files, folders, documents, all strewn anywhere you wanted
- PCs improved on the real world because we could auto organize

Not all metaphors are created equal …

- However, there were some problems
  - Windows made it so that people felt that everything was a file – and on your desk you can move any file anywhere you want – but if you did that in Windows it all went horribly wrong
- It wasn't until Windows XP that Microsoft finally got around to hiding those files from the user – and by that time most of us had figured out the dangers (often by the horrible badness that occurred)

Limits of Metaphors

- Metaphors are fantastic in a lot of ways – but there are problems you need to be aware of when you build an interactive technology:
  - If people think they can apply rules to something, they likely will try – so you need to think about what happens when they try to do something that is forbidden
  - Not all metaphors transfer into the digital setting – Apple made a major departure in their desktop metaphor by putting the trashbin on the desktop – your trash bin goes under the desk! But you couldn't do that digital computing

Returning to Wikipedia …

- What are the key metaphors that are used in Wikipedia?

What about other technologies?

- Think about your smartphone – what are the key things that you can do on your smartphone?
  - What are the metaphors that support some of those things?

Conclusions

- The image of the “dumb user” is one that computing propagates by continuing to design technology badly
- “Dumb users” are usually provoked by the system into making mistakes – this lowers the usability of the system – specifically the effectiveness of the system
- If you want to have a usable system, you need to start with the core concepts, relationships between those concepts and understand how they relate to the mental models of the users
- You can leverage metaphors to try to build on existing skills of the users, but you need to watch out for negative skill transfer

Readings

- Cooper et al.
  - Chapter 2 Implementation Models and Mental Models
  - Chapter 13 Metaphors, Idioms and Affordances
    - Up to page 280