Conceptual Design

Human Computer Interaction

Pearl Pu

Motivation

- Many products designers do not consider the expectations of users
- Difference between engineer vision and user experience
- Conceptual design should be done prior to physical design

What is Conceptual Design

- Useful software should be easily understood
- Achieve compromise between complexity and usability
- => understand the problem space of users and design conceptual model of interactions

Definitions

- The problem space is the set of tasks that software systems can improve (automate) for users
- The conceptual model is a high-level description of how a system is organized and operates to achieve a goal
- A user centered design starts from understanding the problem space and derive the system operators that resolve them

1. Understanding the problem space

- What do you want to create?
- What are your assumptions?
- What are your claims?
- Will it achieve what you hope it will?
- If so, how?

1.1 Analyzing the problem space

- Are there problems with the existing product or user experience?
- Why do you think there are problems?
- How do you think your proposed design ideas might overcome these?
- When designing for a new user experience, how will the proposed design extend or change current ways of doing things?
A simple example

- ATM machine
- Problem space
  - Assumptions:
    - Draw money when bank is closed
    - Machines are more precise than tellers
  - Claim: ATM will be everywhere
- Conceptual model
  - Draw cash, check balance, deposit, pay bills, buy stamps, lottery tickets

1.1.1 A more Complex Example

- The online photo sharing and management applications
  - One of the most popular and successful application is Flickr (http://www.flickr.com/)
  - What were the main assumptions made by developers?

1.1.2 Assumptions and claims

- Assumptions = taking something for granted
  - Able to capitalize on the successful phenomenon of blogging
  - People want to share theirs photos with the rest of the world, in the same way they blog
  - They want to get comments back about their photo collections
- A claim = stating something to be true when it is still open to question
  - «Flickr is almost certainly the best online photo management and sharing application in the world» (Flickr’s website, 2005)

1.2 Exercise

- What would be your assumptions before developing a wine recommender system?
  - The complete description and features of wines are too complicated for the general audience
  - Tastes about wines are relatively stable and it should be easy to learn users’ preferences and make suggestions
  - Able to capitalize on the successful phenomenon of recommender systems (Amazon, Fnac, …)
  - They want to know what to drink according to what they will eat
  - Other assumptions?

1.3 From problem space to design space

- Problem space: discover new wines, get wines delivered 7/24, get wine suggestions based on food
- Conceptual model – search for wine known to users, list top picks, recommend wines based on past purchase history,
2.1 Analogy with software engineering
- Software engineering
  - Think about functionalities
  - Model the application with an UML-like representation
    - Not a description of the source code but of the architecture and functionalities of the system
    - Enables developers to structure their thinking before they start coding
  - Develop and debug
- Interaction design
  - Define the problem space
  - Develop the conceptual model
    - Not a description of the user interface but a structure outlining the concepts and the relationships between them
    - Enables designers to straighten out their thinking before they start laying out their widgets
  - Define UI and evaluate with real users

2.2 Goal of conceptual model
- Help the design team
  - Orient themselves towards asking questions about how the conceptual model will be understood by users
  - Establish a set of common terms they all understand and agree upon
  - Reduce the chance of misunderstandings and confusion arising later on

2.3 Main components of conceptual design
- Metaphors and concepts
- Interaction types
- Theories, models and frameworks

2.3.1 Metaphors and concepts
- Metaphors and analogies are used to convey what a product is for and how to use it for an activity
- Concepts are objects that users are exposed to through the product
  - There can be relations between concepts (one object contains another)
  - There is a mapping between the concepts and the user experience the product is designed to support

2.3.1.1 Example
- A spreadsheet, like MS Excel
  - Analogous to ledger sheet
  - Interactive and computational
  - Easy to understand for users
    - It is just a tool to allow others to work out their ideas and reduce the tedium of repeating the same calculations
    - Capitalized on user’s familiarity with ledger sheets
  - Greatly extending what accountants and others could do
    - Got the computer to perform a range of different calculations and recalculations in response to user input

2.3.1.2 Interface metaphors
- Designed to be similar to a physical entity but also has own properties
  - Ex: Desktop metaphor
- Exploit user’s familiar knowledge, helping them to understand the unfamiliar
- People find it easier to learn and talk about what they are doing at the computer interface in terms familiar to them
2.3.1.3 Advantages and drawbacks

- **Advantages**
  - Makes learning new systems easier
  - Helps users understand the underlying conceptual model
  - Makes the applications accessible to a greater diversity of users

- **Drawbacks**
  - Can constrain designers in the way they conceptualize a problem space
  - Forces users to only understand the system in terms of the metaphor
  - Limits designers’ imagination in coming up with new conceptual models

2.3.2 Interaction types

- **Instructing**
  - Issuing commands using keyboard and function keys and selecting options via menus

- **Conversing**
  - Interacting with the system as if having a conversation
  - Interacting with objects in a virtual or physical space by manipulating them
  - Moving through a virtual environment or a physical space

- **Exploring**
  - Moving through a virtual environment or a physical space

2.3.2.1 Instructing

- Where users instruct a system by telling it what to do
  - Tell the time, print a file, find a photo

- Very common interaction type underlying a range of devices and systems

- A main benefit of instructing is to support quick and efficient interaction

2.3.2.2 Conversing

- Like having a conversation with another human

- Differs from instructing in that it is more like two-way communication, with the system acting like a partner rather than a machine that obeys orders

- Ranges from simple voice recognition menu-driven systems to more complex ’natural language’ dialogues

2.3.2.2 Examples of conversational systems

- Advice-giving systems (recommender systems) and help systems (expert systems)

- A.L.I.C.E. bot (http://alice.pandorabots.com/)

2.3.2.2 Advantages and drawbacks

- **Advantage**
  - Allows users, especially novices, to interact with the system in a familiar way and makes them feel at ease and less scared

- **Drawback**
  - Misunderstandings can arise when the system does not know how to parse what the user says
2.3.2.3 Manipulating

- Exploits users’ knowledge of how they move and manipulate in the physical world
- Virtual objects can be manipulated by moving, selecting, opening, and closing them

2.3.2.3 Direct manipulation

- Introduced by Schneiderman (1983)
- Proposes to design digital objects that can be handled like physical objects
- Assumes that direct manipulation interfaces enable users to feel that they are directly controlling the digital objects
- Ex: physical actions and button pressing instead of issuing commands with complex syntax, rapid reversible actions with immediate feedback on object of interest

2.3.2.3 Advantages and drawbacks

- Advantages
  - Novices can learn the basic functionalities quickly
  - Intermittent users can retain operational concepts over time
  - Users can immediately see if their actions are furthering their goals and if not do something else
  - Users experience less anxiety, gain confidence and mastery
- Drawbacks
  - Not all tasks can be described by objects and not all actions can be done directly
  - Some tasks are better achieved through delegating rather than manipulating (ex: spell checking)
  - Moving a mouse around the screen can be slower than pressing function keys to do same actions

2.3.2.4 Exploring

- Involves users moving through virtual or physical environments
- Examples include:
  - 3D virtual worlds where people navigate using mouse to socialize (Second life, etc.)
  - CAVEs where users navigate by moving whole body, arms, and head
  - Physical context aware worlds, embedded with sensors, that present digital information to users at appropriate places and times

2.3.3 Theories, models and frameworks

- These are instruments used to inform and inspire design
- A theory is a well-substantiated explanation of some aspect of a phenomenon
- A model is a simplification of some aspect of human-computer interaction intended to make it easier for designers to predict and evaluate alternative designs
- A framework is a set of interrelated concepts and/or a set of specific questions
2.3.3 Differences

- Theories tend to be comprehensive, explaining human-computer interactions
- Models tend to simplify some aspect of human-computer interactions
- Frameworks tend to be prescriptive, providing designers with concepts, questions, and principles to consider

Conclusion

- You have to conceptualize interaction in the same way you model the system in software engineering
- It involves to understand the problem space and to develop a conceptual model before designing the user interface

References