Software Construction and User Interface (SE/ComS 319)

Ali Jannesari

Department of Computer Science

Iowa State University, Fall 2018

USER INTERFACES

Outline

- User interfaces
- Design principles for user interfaces
- User interfaces construction

User interface

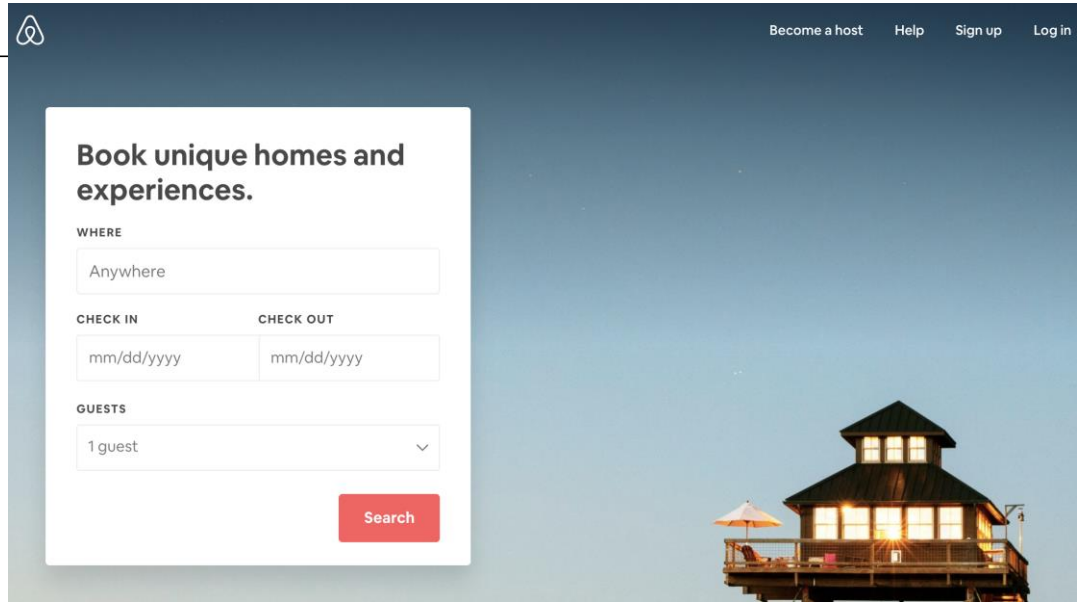
- User interface: Way by which end-users will interact with your software
- Should take into consideration users' expectations, experience and skills
- Bad interface ➔ low usability

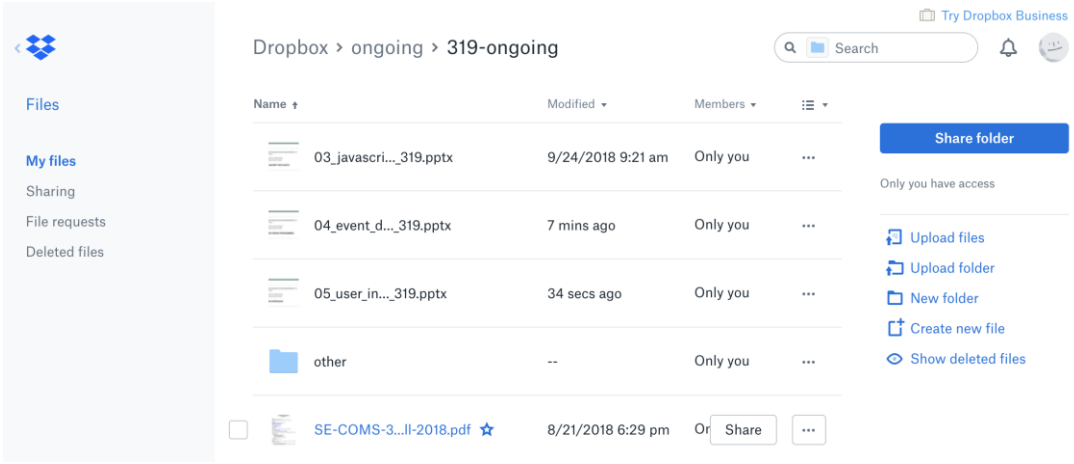
User interface

- Human errors have been correlated to the usability of user-interface (J. Galliers and et al, ACM TCHI).
  - Confirmation dialog box:

User interface

- Examples of excellent UI:
  - AirBnB, DropBox, …





User interface

- Examples of poor UI
  - IBM Lotus Notes
  - Windows 8,…

Use of a visible language

- Layout
- Typography: selection of typefaces and typesetting, including variable width and fixed width
- Color and Texture
- Imagery
- Sequencing
- Sound
- Visual identity: the additional, unique rules that lend overall consistency to a user interface

Use of a visible language (2)

- **Organize**: provide the user with a clear and consistent conceptual structure
  - Consistency, screen layout, relationships and navigability
- **Economize**: do the most with the least amount of cues
  - Simplicity, clarity, distinctiveness, and emphasis
- **Communicate**: match the presentation to the capabilities of the user.
  - In order to communicate successfully keep in balance legibility, readability, typography, symbolism, multiple views, and color/texture.

Design principles for user interfaces

- Design Issues
- Design Process
- Evaluation

UI Design Issues

- Human factors
- Interaction styles (to/from the user)
  - Visualization
  - Error/warnings
  - Color
  - . . .

UI Design Issues – Human Factors

- Limited short-term memory
  - How many items of information can one remember instantaneously?
- Familiarity
  - Use terms and concepts from the domain of the application
- Consistency
  - Similar/comparable operations should be activated in the same way
- Error recovery & guidance
  - Provide meaningful, unambiguous feedback when errors occur

UI Design Issues – Interaction styles

- Obtaining information from the user
- Presenting information to the user
  - Direct Manipulation
  - Menu-based
  - Form-based
  - Natural language
  - Command language

Interaction styles – Input

- Direct manipulation
  - Advantage: Intuitive interaction
  - Disadvantage: Hard to implement, requires visual metaphor
  - Applications: Games, CAD

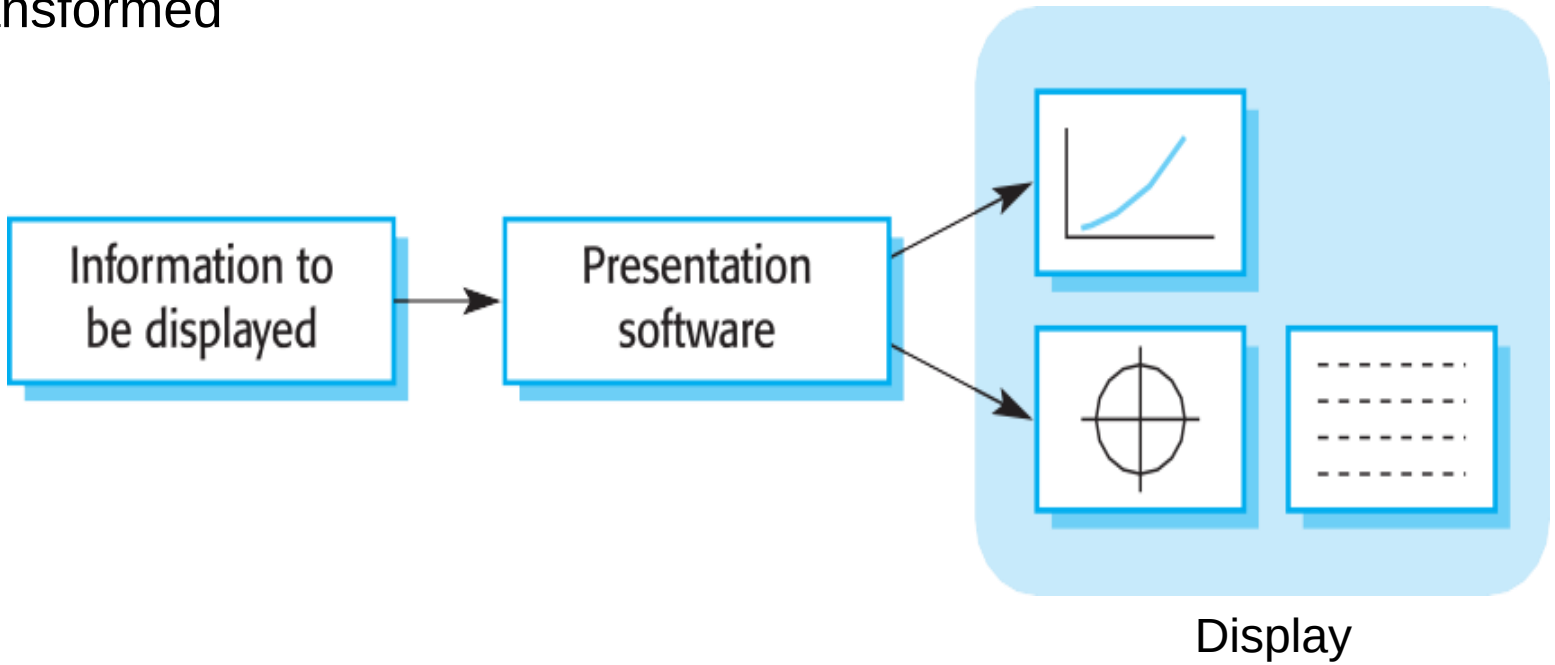
- Menu-based
  - Advantage: Avoids user error
  - Disadvantage: Can be slow and/or complex
  - Applications: Most systems

Interaction styles – Input

- Form-based
  - Advantage: Simple and Checkable
  - Disadvantage: Can be long
  - Applications: e-Commerce
- Natural Language
  - Advantage: easy and natural
  - Disadvantage: Natural language processing (NLP)
  - Applications: Information retrieval systems, apps
- Command-Line Language
  - For all of us: easy to implement, hard to understand/remember all commands

Interaction styles – output

- Direct
- Transformed



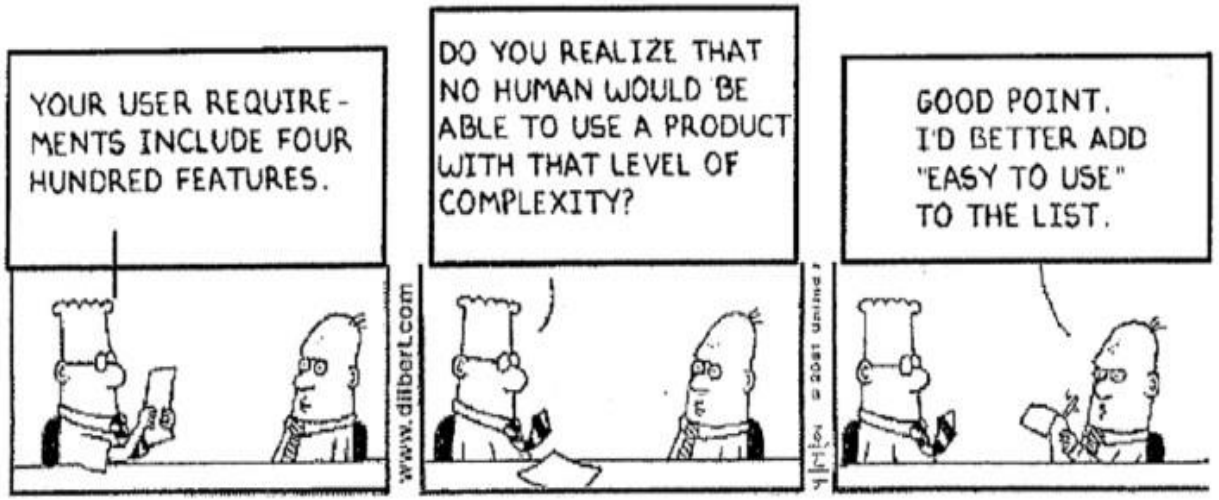
Data Visualization

- Techniques for displaying large amounts of information
- May reveal relationships and/or trends of data
  - Could improve human problem-solving performance
  - Could influence Business Decisions
  - **Wrong inferences!!** (Risks of Visualization)

Design Process of UI

- User Analysis
  - Understand what users will do with the system
- Prototyping
  - Develop (many) prototypes
- Evaluation
  - Experiment with the prototypes

User Analysis



User Analysis

- Ask questions, show examples, explain what can (more importantly cannot) be done, . . .
  - Requirements solicitation
  - Ethnography (Observe the user at work)

- **Tangible** information (feedback) from user:
  - I want to control my computing device using voice commands
  - I want to control my computing device using voice commands and it must only follow my voice commands
  - I want to mind-control my computing device

Prototyping

- Provide users a direct experience with the interface
- Helps in getting users' judgment
- Simple prototypes
  - Paper + pencil (story-boards, scenarios, use-cases, etc.)
  - Digital with dummy buttons
    - e.g., Pencil Project http://pencil.evolus.vn/ (free and open-source GUI prototyping tool)
  - Digital with some functionality (e.g., scripting, visual language, etc.)
- . . .

Evaluation – Usability

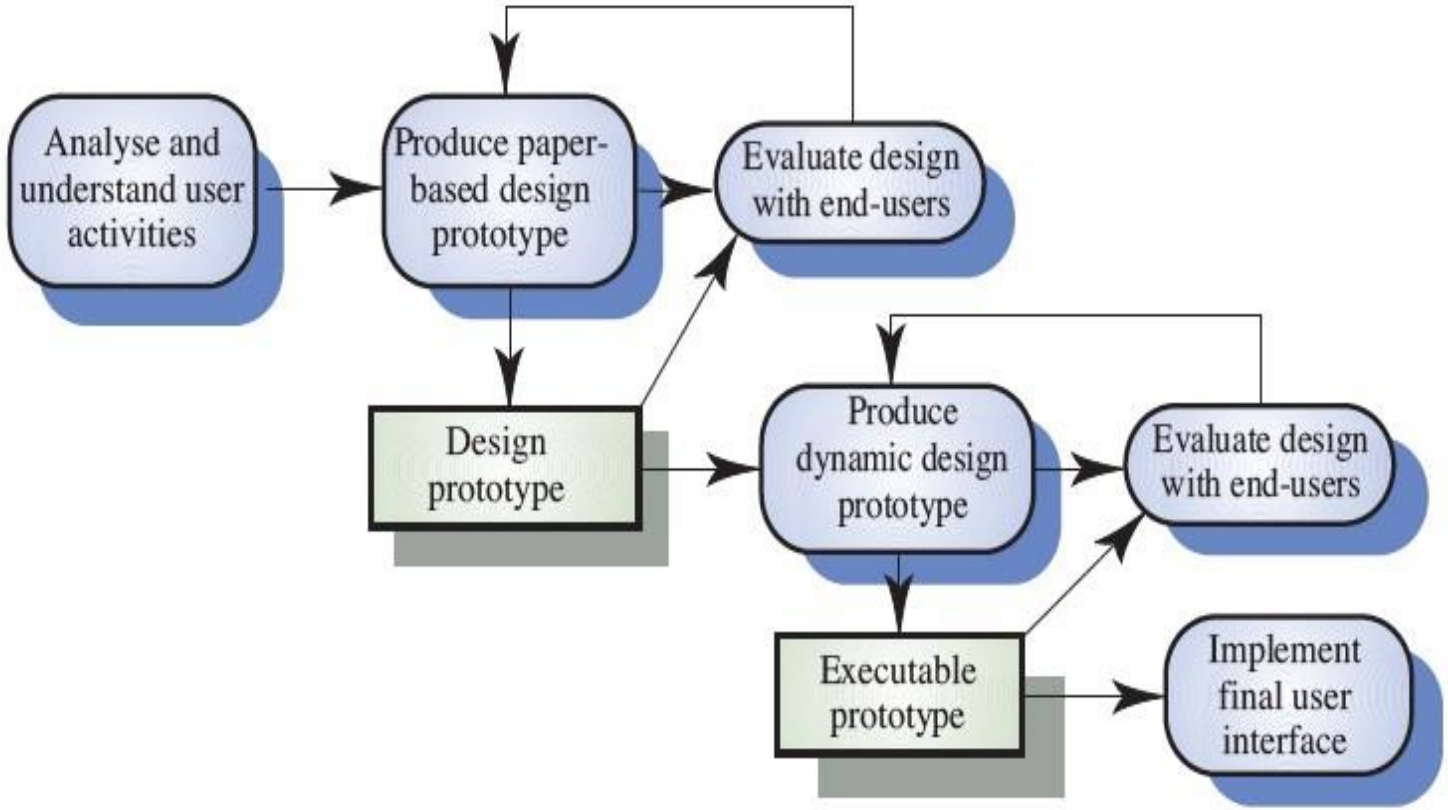
- Conformance to domain-specific vocabulary
- Recognition of options
- Consistency
- Visibility of system status
- Error prevention
- Error information
- Easy recovery methods
- Precise and concise information
- Help and manuals
- Flexibility for experts

Evaluation – Usability

Designing the User Interface (Ben Shneiderman):

1. Strive for consistency.
2. Give shortcuts to the user.
3. Offer informative feedback.
4. Make each interaction with the user yield a result.
5. Offer simple error handling.
6. Permit easy undo of actions.
7. Let the user be in control.
8. Reduce short-term memory load on the user.

Typical UI-Design (UI Development cycle)



User interface principles – Recap

- Keep the users in mind
- Get feedback often
- Prepare multiple (progressively advanced) prototypes

Construction of User Interface (UI)

- UI allows users interact with the data
  - Manage
  - View
  - Modify
- Location of data & type of interaction
  - Stand-alone applications: data hosted on the client
  - Client-server applications
    - Data hosted on the server, user-interface and computations on the client
    - Data hosted on the server, computations on the server, user-interface on the client
    - Data hosted on the server, computations partitioned between client and server (data requested when needed or pre-fetched), user-interface on the client.

Software development process (1)

- System analysis & requirements elicitation
  - Analysis: What are the functional and non-functional requirements of the desired system?
    - Domain model: Relationship of the software with the real-world.
    - Application model: Description of application functionality.
- System design: High-level architecture of the application
  - Relationships: UML and modularization
  - Class diagrams: Organize the data (information hiding, interface specification)
  - Interface design (interactions between software modules)
- Implementation & Testing
- Deployment

Software development process (2)

- Design considers domain & data, modularization and interface
- Distinction between data, components and interface during design
  - Data design ➔ data structures
  - Component (package) design ➔ separation of functionalities
    - Decomposability
    - Composability
    - Understandability (Individuality)
    - Continuity (Extensibility)
    - Protection (Security)
  - Interface design ➔ reduction of communication complexity
    - Example of interfaces in Java: JavaFX, Swing, AWT for GUI development and event-driven programming

Summary

- User interfaces
- Design principles for user interfaces
- User interfaces construction

Literature – User Interfaces

- https://www.interaction-design.org/literature/topics/ui-design
- https://blog.teamtreehouse.com/10-user-interface-design-fundamentals
- http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

- Designing the User Interface (6th Edition) by Ben Shneiderman and Catherine Plaisant