Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

June 24, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter One: Interaction Design

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: https://vincentnofong.com/
- Uni website: https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
- End of Semester: **60%**

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Introduction

Rules

1. Feel free to ask questions in class, unless they are too "personal".
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter One: Interaction Design

Course Instructor: Vincent M. Nofong, Ph.D.

June 24, 2024

What is Human Computer Interaction (HCI)?

- HCI is the study and the practice of usability.
  - It is about understanding and creating software and other technology that people will want to use, will be able to use, and will find effective when used.
- HCI is the study of how people use computer systems to perform certain tasks.
  - HCI tries to provide us with all understanding of the computer and the person using it, so as to make the interaction between them more effective and more enjoyable.

Humans, Computer and Interactions

- The H:
  - Humans are good at: sensing low level stimuli, pattern recognition, inductive reasoning, multiple strategies, adapting "Hard and fuzzy things".
- The C:
  - Computers are good at: counting and measuring, accurate storage and recall, rapid and consistent responses, data processing/calculation, repetitive actions, performance over time, "Simple and sharply defined things".
- The I:
  - Let humans do what humans do best and computers do what computers do best.

Why Care? - Motivation

- HCI is the study of the ways that people use computers - aimed at making computers easier for people to use.
- Is that possible to make computers easier for people to use? Yes!
- It happens when people who design computers and software keep in mind that they are designing for other people.

Different Design Needs

The three broad categories of computer user:

- **Expert users** with detailed knowledge of that particular system.
- **Occasional users** who know well how to perform the tasks they need to perform frequently.
- **Novices** who have never used the system before.

Note: Users may well be novices at one computer application but experts at another one, so users will belong to different categories for particular computer systems.

What is Interaction Design?

- "Designing interactive products to support the way people communicate and interact in their everyday and working lives". - Preece, Sharp and Rogers (2019)
- "The design of spaces for human communication and interaction". - Winograd (1997)

Goals of Interaction Design:

- Develop usable products - easy to learn, effective to use, and provide an enjoyable experience.
- Involve people and users in the design process.
- Consider what people are good and bad at.
- Consider what might help people with the way they currently do things.
- Think through what might provide quality experiences.
- Consider a person's privacy concerns if data is being collected about them.
- Use people-centered techniques during the design process.

HCI and Interaction Design

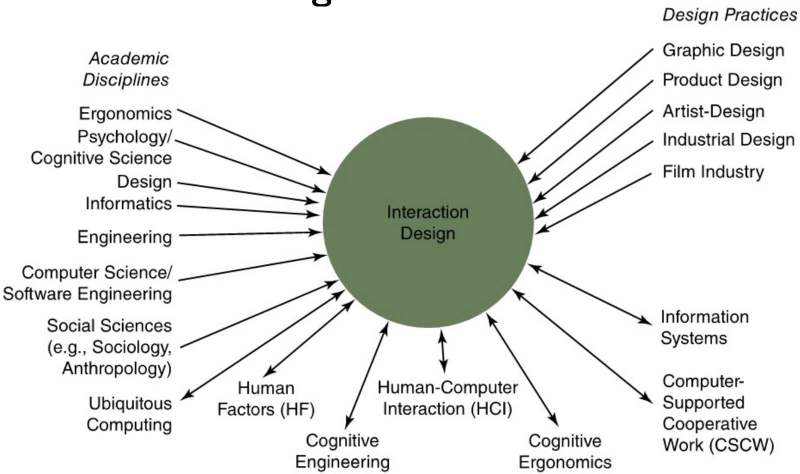


Figure 1: Interdisciplinary Fields in Interactive Design

Interaction Design

Working in multidisciplinary teams

- Many people from different backgrounds involved
- Different perspectives and ways of seeing and talking about things
- **Benefit:** more ideas and designs are generated
- **Disadvantage:** difficult to communicate and progress forward the designs being created

Interaction Design

Categories of Professionals in Interactive Design Business

- **Interaction Designers** - people involved in the design of all the interactive aspects of a product.
- **Usability Engineers** - people who focus on evaluating products, using usability methods and principles.
- **Web designers** - people who develop and create the visual design of websites, such as layouts
- **Information Architects** - people who come up with ideas of how to plan and structure interactive products.
- **User Experience Designers (UX)** - people who do all the above but who may also carry out field studies to inform the design of products

What to design - things to consider

- Need to take into account:
  - Who the users are
  - What activities are being carried out
  - Where the interaction is taking place
- Need to optimize the interactions users have with a product:
  - So that they match the users' activities and needs
  - Bad design example: `http://bolden.nl/`
    - Is this a clever design? Yes, definitely.
    - But is this bad design? Absolutely!
    - This is a great example of designing for the designer, rather than the user:

The User Experience

The user experience refers to:

- How a product behaves and is used by people in the real world.
- The way people feel about it the product
- Their pleasure and satisfaction when using the product, looking at it, holding it, and opening or closing it.

Note: You cannot design a user experience, you can only design for a user experience.

The User Experience: iPod User Experience Success

Why was the iPod a success?

- Designers ensured quality user experience from the start.
- Designers ensured it is simple, elegant, distinct brand, pleasurable, must have fashion item, catchy name, cool, etc.

Core characteristics of interaction design

- Users should be involved through the development of the project
- Specific usability and user experience goals need to be identified, clearly documented and agreed at the beginning of the project
- Iteration is needed through the core activities

Core characteristics of interaction design: Why?

To help designers:

- understand how to design interactive products that fit with what people want, need and may desire
- appreciate that one size does not fit all e.g., teenagers are very different to grown-ups
- identify any incorrect assumptions they may have about particular user groups e.g., not all old people want or need big fonts
- be aware of both people's sensitivities and their capabilities

Accessibility and Inclusiveness

- **Accessibility:** the extent to which an interactive product is accessible by as many people as possible
  - Focus is on people with disabilities; for instance, those using android OS or apple voice-over.
- **Inclusiveness:** making products and services that accommodate the widest possible number of people .
  - For example, smartphones designed for all and made available to everyone regardless of their disability, education, age, or income

Disabilities

- Whether someone is disabled changes over time with age, or recovery from an accident
- The severity and impact of an impairment can vary over the course of a day or in different environmental conditions
- Disabilities can result because technologies are designed to necessitate a certain type of interaction that is impossible for someone with an impairment

Understanding disability (1/2)

Disabilities can be classified as:

- Sensory impairment (such as loss of vision or hearing)
- Physical impairment (having loss of functions to one or more parts of the body after a stroke or spinal cord injury)
- Cognitive (including learning impairment or loss of memory/cognitive function due to old age)

Understanding disability (2/2)

Each disability type can be further defined in terms of capability:

- For example, someone might have only peripheral vision, be color blind, or have no light perception

Impairment can be categorized:

- Permanent (for instance, long-term wheelchair user)
- Temporary (that is, after an accident or illness)
- Situational (for example, a noisy environment means that a person can't hear)

Usability Goals

- Effective to use (doing the right thing)
  - How good a product is at doing what it is supposed to do
- Efficient to use (doing things right)
  - Product supports users carrying out their tasks efficiently.
- Safety: Product is safe to use
- Have good utility
  - Product provides a right kind of a functionality so users can do what they need or want to do
- Learnability: Product is easy to learn
- Memorability
  - Easy to remember how to use product
- Product is enjoyable to use

User experience goals (1/2)

Desirable aspects

- satisfying
- enjoyable
- engaging
- pleasurable
- exciting
- entertaining
- helpful
- motivating
- challenging
- enhancing sociability
- supporting creativity
- cognitively stimulating
- fun
- provocative
- surprising
- rewarding
- emotionally fulfilling

User experience goals (2/2)

Undesirable aspects

- boring
- frustrating
- annoying
- childish
- intrusive
- unpleasant
- patronizing
- cutesy
- gimmicky
- creepy
- deceptive
- making one feel stupid
- making one feel guilty

Design principles

- Generalizable abstractions for thinking about different aspects of design
- The do's and don'ts of interaction design
- What to provide and what not to provide at the interface
- Derived from a mix of theory-based knowledge, experience and common-sense
- Main principles: Visibility, Feedback, Constraints, Consistency, Affordance

Visibility Principle - Poor Interface

Figure 2: Elevator Control Panel

- How does it work?
- Push a button for the floor you want?
- Nothing happens. Push any other button? Still nothing. What do you need to do?
- It is not visible as to what to do!

Visibility Principle - Improving on a poor Interface

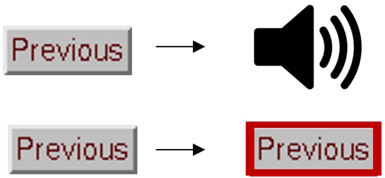
Figure 3: Elevator Control Panel

With this elevator, you need to insert your room card in the slot by the buttons to get the elevator to work!

- How would you make this action more visible?
  - Make the card reader more obvious
  - Provide an auditory message that says what to do (which language?)
  - Provide a big label next to the card reader that flashes when someone enters
  - Make relevant parts visible
  - Make what has to be done obvious

Feedback Principle

- Sending information back to the user about what has been done
- Includes sound, highlighting, animation, and a combinations of these.
- For example, when screen button is clicked, it provides sound or red highlight feedback:



Constraints Principle

- Restricting the possible actions that can be performed
- Helps prevent user from selecting incorrect options
- Physical objects can be designed to constrain things
- E.g. only one way you can insert a key into a lock

Consistency Principle

- Design interfaces to have similar operations and use similar elements for similar tasks.
- For example, always use Ctrl key plus first initial of the command for an operation: Ctrl+c, Ctrl+s, Ctrl+o
- The main benefit is that consistent interfaces are easier to learn and use

Consistency Principle -When consistency breaks down

- What happens if there is more than one command starting with the same letter?
    - For example, save, spelling, select, style
- You have to find other initials or combinations of keys, thereby breaking the consistency rule
    - For example, Ctrl+s, Ctrl+Sp, Ctrl+shift+l
- Increases learning burden on user, making them more prone to errors

Consistency Principle -Internal and External Consistency

Internal Consistency

- Refers to designing operations to behave the same within an application
- Difficult to achieve with complex interfaces

External Consistency

- Refers to designing operations, interfaces, and so on to be the same across applications and devices
- Very rarely the case, based on different designer's preference

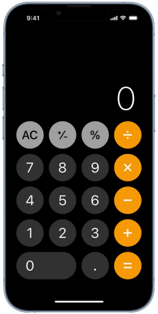
Interaction Design

Consistency Principle -A Case of External Consistency Keypad numbers layout

Phone dial pad

TV Remote

Computer Numeric Keypad



Phone Calculator

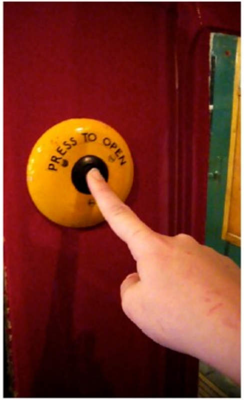
Affordance Principle

- An invitation to action
- Refers to an attribute of an object that allows people to know how to use it
- E.g. a mouse button invites pushing, a door handle affords pulling.

Affordance Principle - Virtual Affordances - Examples

Switch suggests toggling

knob suggests turning



button suggests pressing

slot suggests inserting, handle suggests turning

Affordance Principle - Virtual Affordances

- How do the following screen objects afford?
- What if you were a novice user? Would you know what to do with them?

Design Principles - Software User Interfaces

1. The Structure Principle
2. The Simplicity Principle
3. The Feedback Principle
4. The Visibility Principle
5. The Tolerance Principle
6. The Reuse Principle

Design Principles - Software User Interfaces

The Structure Principle (1/2)

- The user interface design should have an organized and well-built structure.
- The design and model should be apparent and recognizable.
- The design should be user-friendly, ensuring ease of use.
- Information should be simplified and separated as much as possible to enhance the user experience.

Design Principles - Software User Interfaces

The Structure Principle (2/2)

Techniques of UI Design that support the principle:

- Navigation between major interface screens should be simple for seamless interaction.
- Grouping objects effectively helps in organizing and presenting information logically.
- The design should be intuitive, allowing users to easily understand and navigate the interface.

Design Principles - Software User Interfaces

The Simplicity Principle (1/2)

- Common and repetitive tasks should be a great experience within the user interface design.
- The design should communicate easily with the user and provide shortcuts or hotkeys for related and/or longer procedures.

Design Principles - Software User Interfaces

The Simplicity Principle (2/2)

Techniques of UI Design that support the principle:

- Consistency in design elements and interactions helps users navigate and interact seamlessly.
- Navigation within the screen should be simple to ensure easy access to different functionalities.
- Effective alignment of fields and elements enhances visual clarity and improves usability.

Design Principles - Software User Interfaces

The Feedback Principle (1/2)

- The user interface design should provide clear and informative feedback to users regarding actions, interpretations, state changes, conditions, errors, or exceptions.
- Language used should be familiar, concise, and unambiguous.

Design Principles - Software User Interfaces

The Feedback Principle (2/2)

Techniques of UI Design that support the principle:

- Explanation of rules helps users understand how the interface functions.
- Anticipating user mistakes and providing guidance or corrective measures.
- Design should be intuitive, allowing users to easily grasp and navigate the interface.

Design Principles - Software User Interfaces

The Visibility Principle (1/2)

- Users should be able to easily locate desired information or options.
- Providing fewer choices helps users make decisions effectively, as too many options can lead to confusion, prolonged decision-making, or incorrect choices.
- UI designers should avoid including irrelevant, unnecessary, or useless information.

Design Principles - Software User Interfaces

The Visibility Principle (2/2)

Techniques of UI Design that support the principle:

- Avoid creating cluttered or busy interfaces that overwhelm users.
- Effectively group objects to enhance visual organization and clarity.
- When reviewing other applications, consider their design choices critically and selectively apply relevant elements.

Design Principles - Software User Interfaces

The Tolerance Principle (1/2)

- UI design should be forgiving and tolerant of user mistakes.
- Design should allow users to easily undo and redo actions to recover from errors.
- Providing helpful recommendations can guide users towards making the right choices.
- A good UI design should accommodate varied input methods and tolerate spelling variations.

Design Principles - Software User Interfaces

The Tolerance Principle (2/2)

Techniques of UI Design that support the principle:

- Anticipate and expect users to make mistakes, and provide mechanisms for easy error correction.
- Design should be intuitive, allowing users to easily understand and navigate the interface.
- Navigation between major interface screens should be simple, facilitating user interactions and tasks.

Design Principles - Software User Interfaces

The Reuse Principle (1/2)

- Consistency in UI design reduces cognitive load and enhances user experience.
- Applying the same elements, layout, actions, and behavior across the user interface or a series of designs promotes familiarity and reduces the need for users to rethink or remember.

Design Principles - Software User Interfaces

The Reuse Principle (2/2)

Techniques of UI Design that support the principle:

- Maintain consistency throughout the UI design, ensuring that elements and interactions follow established patterns.
- Set project standards to establish guidelines and ensure consistency across different UI designs.
- Clearly explain the rules and guidelines to users, helping them understand and navigate the interface consistently.

Design Principles - Software User Interfaces

The Reuse Principle

- Adobe Products (Photoshop and Lightroom)
- Microsoft Products (Word, Excel, PowerPoint)

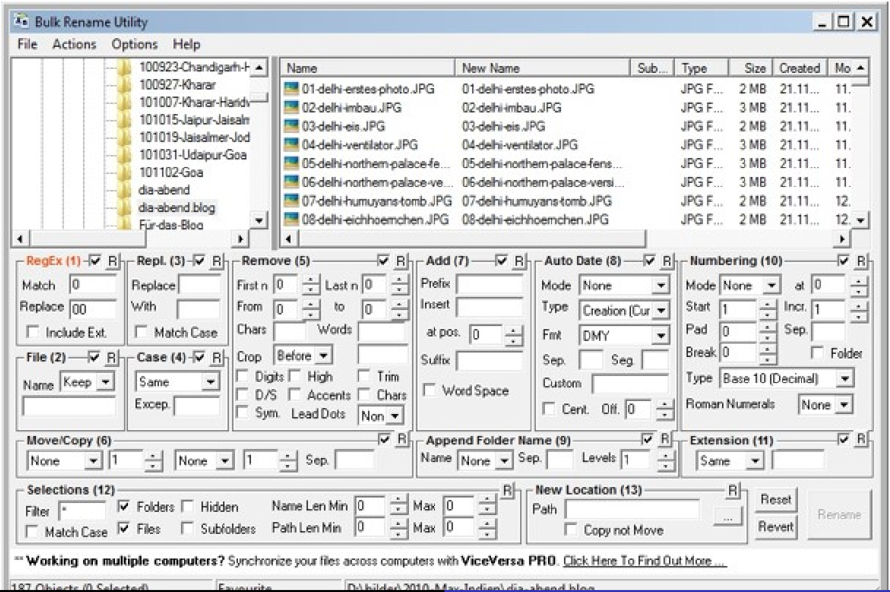
Design Principles - Software User Interfaces: Activity

Which will you recommend? Why?

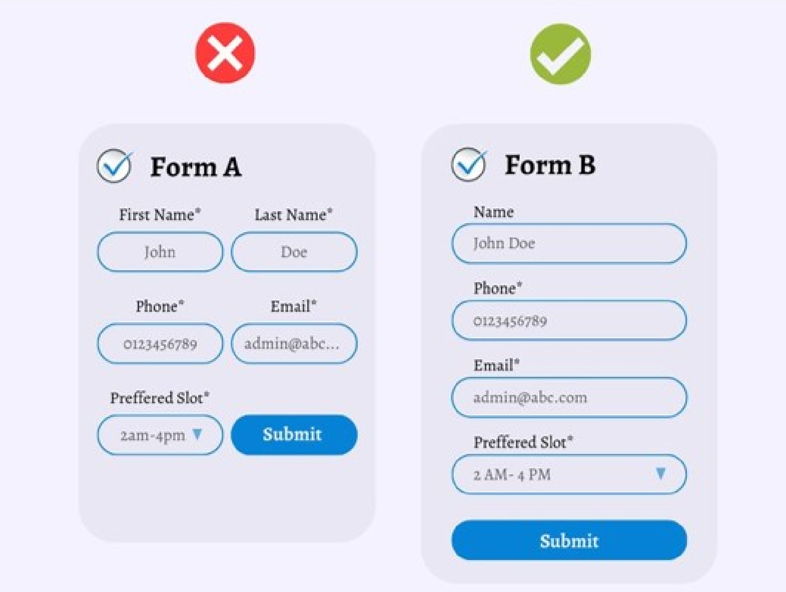
A

B

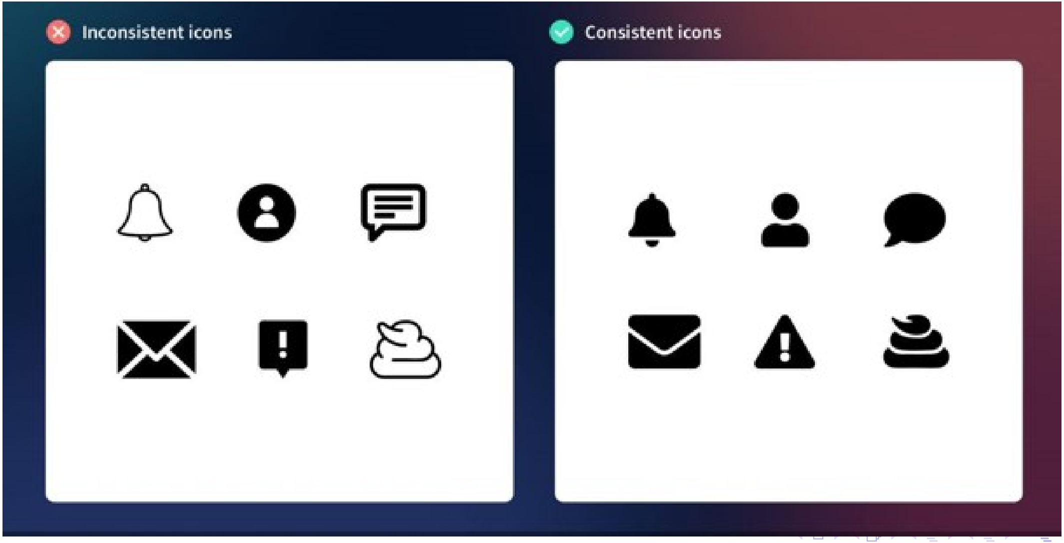
Which Design Principles are not followed? - Be better than this



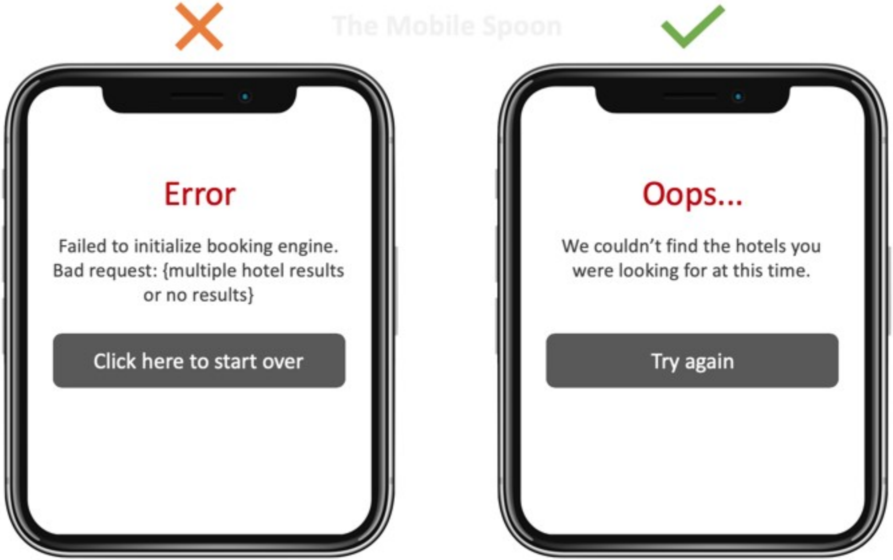
Good vs Bad UI



Good vs Bad UI



Good vs Bad UI



Interaction Design

Summary

- Interaction design is concerned with designing interactive products to support the way people communicate and interact in their everyday and working lives
- It is concerned with how to create quality user experiences
- It requires taking into account a number of interdependent factors, including context of use, type of activities, cultural differences, and user groups
- It is multidisciplinary, involving many inputs from wide-reaching disciplines and fields

Create designs with a user-centric approach, focusing on the needs of your audience rather than your own preferences.

Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

June 24, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: https://vincentnofong.com/
- Uni website: https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Rules

1. Feel free to ask questions in class, unless they are too “personal”.
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Two: Establishing Requirements

Course Instructor: Vincent M. Nofong, Ph.D.

June 24, 2024

Establishing Requirements

Group Assignment

- Group A should divide themselves into 4 groups
- Group B should divide themselves into 4 groups
- I want the list of members on each group by our next meeting, the application they are developing, and the user category.

Establishing Requirements

Group Assignment: Tasks

Question 1: Design and implement a Health and Wellness App:

1. For young children and teenagers - 18 years and below
2. For elderly people - 50 years and above

Question 2: Design and implement an Educational Platform App:

1. For young children and teenagers - 18 years and below
2. For elderly people - 50 years and above

Establishing Requirements

Group Assignment: Tasks

Question 3 - Design and implement a Digital Library App:

1. For young children and teenagers - 18 years and below
2. For elderly people - 50 years and above

Question 4: Design and implement an Entertainment Hub App:

1. For young children and teenagers - 18 years and below
2. For elderly people - 50 years and above

One group per user category.

First Come, First Served, aka, fight among yourselves to decide which question your group picks

Establishing Requirements

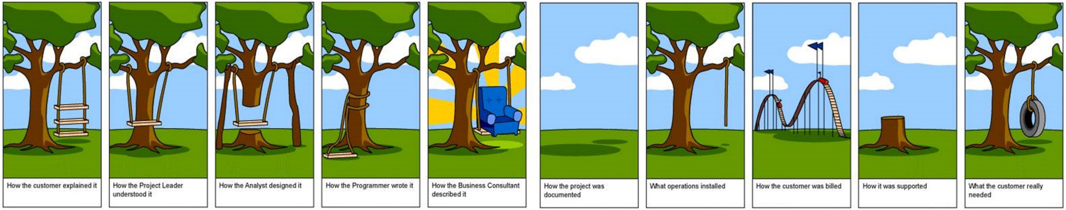
What are Requirements?

- Requirements are statements that define what a product is expected to do or how it will perform.
- They come in different forms and levels of abstraction.
- User stories are commonly used in agile development contexts to capture requirements.
- User stories follow a specific format:
  - *"As a <role>, I want <behavior> so that <benefit>."*

What are Requirements?

- Example User Stories for a Travel Organizer:
  - As a <traveler>, I want to <save my favorite airline for all my flights> so that <I will be able to collect air miles>.
  - As a <travel agent>, I want <my special discount rates to be displayed to me> so that <I can offer my clients competitive rates>.
- These user stories provide specific examples of how requirements can be expressed for a travel organizer, focusing on the needs and desired benefits of different user roles.

Why are Requirements Important?



- Miscommunication is most commonly observed during the requirements activity stage.
- Ensuring accurate and clear requirements is of utmost importance.

Establishing Requirements

Functional vs. Non-functional Requirements

Functional Requirements:

- Define the specific behaviours and functionalities that the system should exhibit.
- They focus on the specific actions and outputs of the system.

Non-functional Requirements:

- Elaborate on the performance characteristics or constraints that the system should adhere to.
- They provide criteria for evaluating aspects such as performance, security, usability, and reliability.

Functional vs. Non-functional Requirements

Functional Requirements:

- Example: The system must send an email whenever a certain condition is met, such as an order being placed or a customer signing up.

Non-functional Requirements:

- Example: Emails should be sent with a latency of no greater than 12 hours from the corresponding activity.

System Stakeholders

- System stakeholders are individuals or organizations who are impacted by the system and have a legitimate interest in its development, implementation, or usage.
- **Stakeholder Types**:
  - End Users - directly interact with the system, utilizing its functionalities to perform tasks or achieve specific goals
  - System Managers - responsible for overseeing the operation, maintenance, and administration of the system
  - System Owners - have ownership or financial responsibility for the system
  - External Stakeholders - entities external to the system who are influenced by or have an interest in the system (customers, regulatory bodies, partners or general public)

Problems of Requirements Elicitation (1/2)

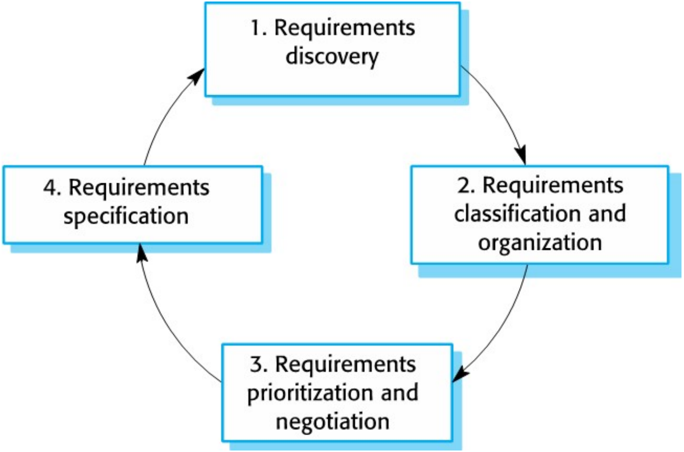
- Stakeholders' Unclear Understanding
  - Stakeholders may have difficulty articulating their needs and desires accurately, leading to a lack of clarity in requirements.
- Stakeholders' Terminology
  - Stakeholders may express their requirements using their own jargon or language, making it challenging to translate their needs into actionable system requirements.
- Conflicting Requirements
  - Different stakeholders may have diverse and sometimes conflicting requirements, creating challenges in prioritizing and reconciling these conflicting demands.

Establishing Requirements

Problems of Requirements Elicitation (2/2)

- Organizational and Political Influences
  - Organizational and political factors can influence the system requirements, leading to biases and preferences that may not align with the optimal solution.
- Changing Requirements
  - Requirements may evolve and change throughout the analysis process.
  - New stakeholders may emerge, and the business environment may undergo transformations, necessitating adjustments to the initial set of requirements.

The Requirements Elicitation and Analysis Process



Establishing Requirements

The Requirements Elicitation and Analysis Process

Process Activities (1/2)

- Step 1 - Requirements Discovery:
  - Engaging with stakeholders to uncover their requirements, including both functional and domain-specific requirements.
  - This stage involves active communication and information gathering.
- Step 2 - Requirements Classification and Organization:
  - Grouping and organizing related requirements into coherent clusters to facilitate a structured and systematic understanding of the overall system needs.
  - This step aids in identifying dependencies and relationships between different requirements.

Establishing Requirements

The Requirements Elicitation and Analysis Process
Process Activities (2/2)

- Step 3 - Prioritization and Negotiation:
  - Assigning priority levels to requirements based on their importance and impact on the system's success.
  - Resolving conflicts and discrepancies between different stakeholder requirements through negotiation and consensus-building.
- Step 4 - Requirements Specification:
  - Documenting the identified requirements in a clear, concise, and unambiguous manner.
  - These specifications serve as input for the subsequent iterations of the development process, ensuring that the requirements are well-documented for further analysis and implementation.

Establishing Requirements

- The process of establishing requirements involves determining what users want and need for a successful system implementation.
  - What do users want? What do users 'need'?
- Requirements often require clarification, refinement, completion, and sometimes even re-scoping to ensure they accurately represent user expectations.
- Input: Requirement document or other relevant sources of information
- Output: Stable requirements that effectively capture user needs and system expectations

Establishing Requirements

Why "Establish" Requirements?

- Establishing requirements is crucial because they arise from a deep understanding of user needs, ensuring that the system aligns with their goals and objectives.
- By establishing requirements, we can justify their relevance and establish clear relationships with relevant data and user insights.
- This process enables us to gather comprehensive and accurate requirements that serve as a foundation for designing and developing a successful system.

Categories of Requirements (1/5)

- Functional Requirements:
  - Statements that define the services the system should provide, its expected reactions to specific inputs, and desired behavior in various situations.
  - May also include statements about what the system should not do.
- Non-Functional Requirements;:
  - Constraints placed on the system's services or functions, such as timing limitations, development process requirements, and adherence to standards.
  - Typically apply to the system as a whole, rather than individual features or services.

Categories of Requirements (2/5)

- Environment or Context of Use
  - **Physical Environment**: consider factors such as dust, noise, vibration, lighting, heat, humidity, etc. E.g., in a hospital setting, these factors may impact the design and functionality of the system.
  - **Social Environment**: Address requirements related to collaboration, coordination, data sharing, distributed systems, synchronous or asynchronous communication, and privacy considerations.
  - **Support**: Specify requirements for user support, communications structure, infrastructure, and availability of training to ensure smooth operation and user satisfaction.
  - **Technical Environment**: Identify the technologies on which the system will run or with which it needs to be compatible, ensuring compatibility and interoperability with existing systems or platforms.

Categories of Requirements (3/5)

- User Characteristics:
  - Consider the characteristics of the users who will interact with the system.
  - Factors to consider include educational background, personal circumstances, abilities, and skills.
- User Profiles:
  - Develop user profiles that capture relevant information about the users, including their roles, preferences, and specific needs.
  - User profiles help guide the design and customization of the system to better align with the diverse user characteristics and requirements.

Establishing Requirements

Categories of Requirements (4/5)

- System Use:
  - Identify the different user profiles based on their experience and frequency of system use.
  - **Novice Users**: These users may require prompts, constraints, and clear instructions to navigate the system effectively.
  - **Expert Users**: These users may value flexibility and desire greater access and control over the system.
  - **Frequent Users**: These users may benefit from shortcuts or efficient ways to accomplish tasks quickly.
  - **Casual/Infrequent Users**: These users may require clear menu paths and intuitive navigation to facilitate ease of use.

Establishing Requirements

Categories of Requirements (5/5)

- Domain Requirements:
  - Constraints imposed on the system due to the specific domain of operation it belongs to.

Data Gathering Techniques for Requirements

- Interviews:
  - Involve direct conversations with stakeholders to gather requirements.
  - Props such as sample scenarios or prototypes can be utilized to facilitate discussions and enhance understanding.
  - Interviews are effective for exploring various issues and allowing development team members to establish a connection with stakeholders.
- Focus Groups:
  - Entail group interviews with multiple stakeholders simultaneously.
  - They are beneficial for obtaining a consensus view on requirements or identifying areas of conflict.
  - However, it is important to be cautious of dominant individuals who may influence the group dynamics.

Establishing Requirements

Data Gathering Techniques for Requirements

- Questionnaires:
  - Questionnaires are frequently used alongside other data gathering techniques.
  - They provide the opportunity to collect both quantitative and qualitative data.
  - Questionnaires are particularly useful when seeking specific responses from a large and geographically dispersed group of people.
- Researching Similar Products:
  - Conducting research on similar products in the market can help prompt and inspire requirements.
  - By analyzing existing products, designers can gain insights into industry trends, best practices, and potential functionalities.

Establishing Requirements

Data Gathering Techniques for Requirements

- Direct Observation:
  - Direct observation involves actively observing stakeholders while they perform their tasks.
  - It provides valuable insights into the nature and context of the tasks being performed.
  - Direct observation is particularly useful for gaining a deep understanding of stakeholders' behaviors and needs.
- Indirect Observation:
  - Indirect observation, such as task logging, is not commonly used in the requirements gathering process.
  - It involves recording and documenting stakeholders' current tasks and activities.
  - Indirect observation can be beneficial for capturing a comprehensive view of stakeholders' tasks and workflows.

Data Gathering Techniques for Requirements

- Studying Documentation:
  - Studying documentation involves reviewing manuals, procedures, and rules.
  - It provides valuable data on the steps involved in specific activities and any regulations or guidelines governing those tasks.
  - Documentation is a good source for understanding legislation and obtaining background information.
  - Documentation should not be used in isolation but in conjunction with other data gathering techniques.

Establishing Requirements

Bringing Requirements to Life

- Scenarios and Personas:
  - These are informal narrative stories that depict specific situations or interactions.
  - They are simple, natural, and personalized, focusing on individual experiences rather than generalizations.
  - Scenarios help in understanding the context, goals, and challenges of users' interactions with a system.
- Use Cases:
  - They assume direct interaction with a system and require a detailed understanding of the interaction.
  - They outline specific actions, inputs, and expected outcomes of a user's interaction with the system.
  - Use cases are beneficial for capturing system behavior and identifying key functionalities.

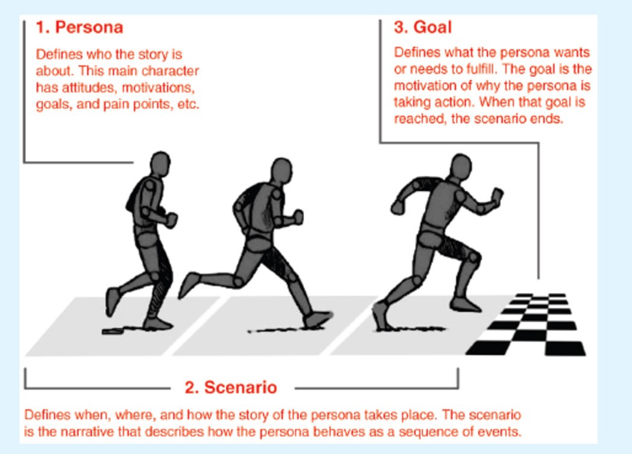
Bringing Requirements to Life

- Essential Use Cases:
  - Essential use cases provide a higher-level abstraction, focusing on the core functionalities of the system.
  - They abstract away from implementation details and specific interactions.
  - Essential use cases help in capturing the essential requirements and system capabilities without making assumptions about the specific details of the interaction.

Establishing Requirements

Bringing Requirements to Life

Scenarios and Personas - Relationship



Establishing Requirements

Users' Capabilities and Variations

Humans vary in many dimensions:

- size of hands may affect the size and positioning of input buttons
- motor abilities may affect the suitability of certain input and output devices
- height if designing a physical kiosk
- strength - a child's toy requires little strength to operate, but greater strength to change batteries
- disabilities (e.g. sight, hearing, dexterity)

The diverse capabilities and variations among users should be considered in establishing requirements

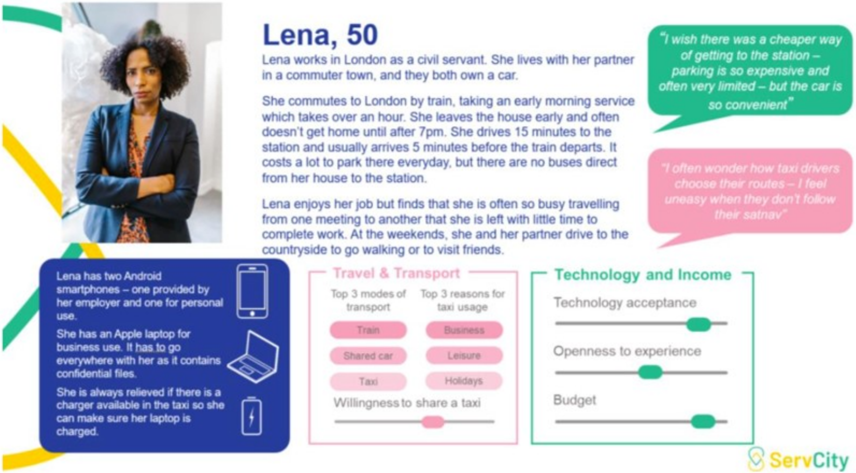
Establishing Requirements

Creating Personas: Bringing Users to Life

- Personas capture a set of user characteristics, representing fictional but synthesized profiles based on real users.
- They provide a human-centered approach to design by bringing users to life with names, characteristics, goals, and personal backgrounds.
- Developing a small set of personas helps designers gain a deeper understanding of user needs and preferences.

By leveraging personas, designers can better empathize with users, align design decisions with user needs, and create more user-centered and engaging experiences.

Creating Personas: Example Persona



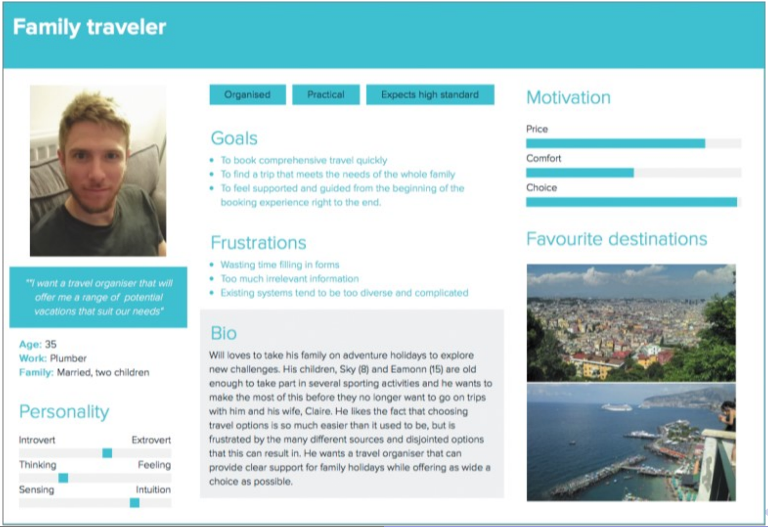
Example Scenario for Group Travel Organizer

Example 1

The Thomson family enjoy outdoor activities and want to try their hand at sailing this year. There are four family members: Sky (8 years old), Eamonn (12 years old), Claire (32), and Will (35). One evening after dinner they decide to start exploring the possibilities. They want to discuss the options together but Claire has to visit her elderly mother so will be joining the conversation from her mother's house down the road. As a starting point, Will enters an idea they had been discussing over dinner – a sailing trip for four novices in the Mediterranean. The system supports users to log on from different locations and use different devices so that all members of the family can interact easily and comfortably with it wherever they are. The system's initial suggestion is a flotilla, where several crews (with various levels of experience) sail together on separate boats. Sky and Eamonn aren't very happy at the idea of going on vacation with a group of other people, even though the Thomson's would have their own boat. The travel organizer shows them descriptions of flotillas from other children their ages and they are all very positive, so eventually, everyone agrees to explore flotilla opportunities. Will confirms this recommendation and asks for detailed options. As it's getting late, he asks for the details to be saved so everyone can consider them tomorrow. The travel organizer messages them a summary of the different options available.

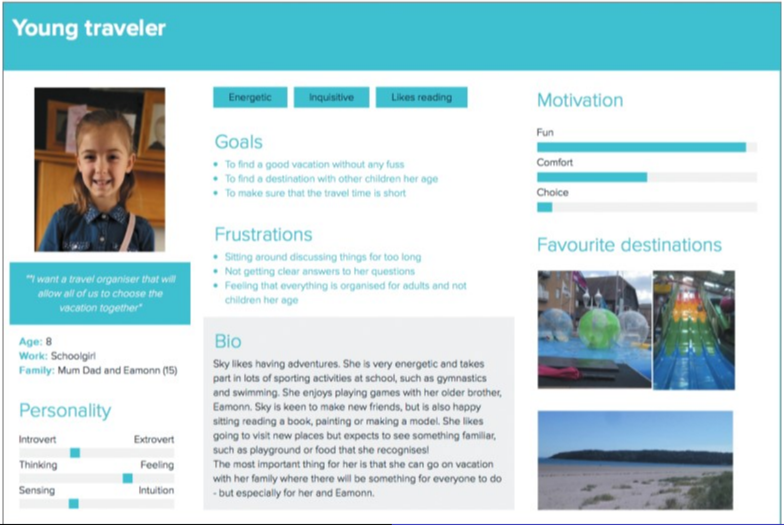
Establishing Requirements

Creating Personas: Example Persona Travel Organizer (1/2)



Establishing Requirements

Creating Personas: Example Persona Travel Organizer(2/2)



Use Cases

- Use cases are a valuable tool for capturing functional requirements and depicting interactions with a system.
- They can be used in the design process or as a means to capture requirements.
- Use cases provide step-by-step descriptions of interactions between users and the system.
- They outline the specific actions, inputs, and expected outcomes of each interaction.

Use Cases Styles

1. Essential Use Cases: These focus on the division of tasks without delving into implementation details.
2. Use Case with Normal and Alternative Courses: This style provides more detailed descriptions, including alternative paths and exceptional scenarios.

Benefits of Use Cases

- Use cases facilitate a clear understanding of the system's functionality and how it interacts with users.
- They help in capturing requirements in a structured and comprehensive manner.
- Use cases can be used as a foundation for design decisions and as a reference throughout the development process.

Establishing Requirements

Case Study: Use Case for Travel Organizer (1/2)

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country's name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.

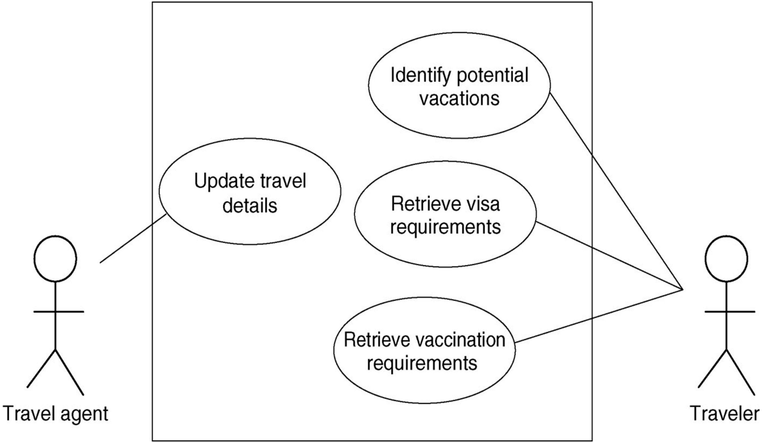
Case Study: Use Case for Travel Organizer (2/2)

8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.
11. The user chooses to print the requirements.

Case Study: Alternate Courses for Travel Organizer

6. If the country name is invalid:
    1. The system displays an error message.
    2. The system returns to step 3.
8. If the nationality is invalid:
    1. The system displays an error message.
    2. The system returns to step 6.
9. If no information about visa requirements is found:
    1. The system displays a suitable message.
    2. The system returns to step 1.

Case Study: Use Case for Travel Organizer



Is the Use Case complete? What are the issues?

Establishing Requirements

Case Study: Essential (Business) Use Case for Travel Organizer

retrieve Visa

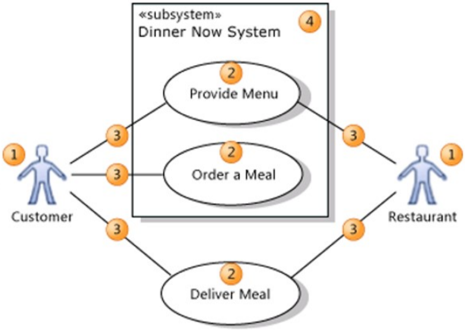
USER INTENTION	SYSTEM RESPONSIBILITY
find visa requirements	request destination and nationality
supply required information	obtain appropriate visa info
obtain copy of visa info	offer info in different formats
choose suitable format	provide info in chosen format

Use Case Modeling

- Use cases were initially developed to facilitate requirements elicitation and are now an integral part of the Unified Modeling Language (UML).
- Each use case represents a distinct task that involves external interactions with a system.
- *Actors* in a use case can be individuals or other systems interacting with the system under consideration.
- Use cases are depicted diagrammatically to provide an overview of the interactions, and they are also described in more detailed textual form.

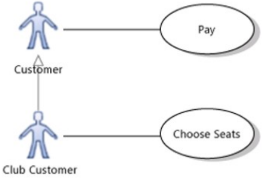
Use Case Modeling Basics (1/5)

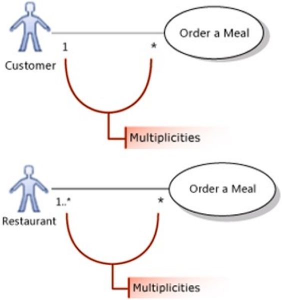
- An *actor* (1) is
  - a class of person, organization, device, or external software component that interacts with your system. Example actors are **Customer**, **Restaurant**, **Temperature Sensor**, **Credit Card Authorizer.**
- A *use case* (2)
  - represents the actions that are performed by one or more actors in the pursuit of a particular goal. Example use cases are **Order Meal**, **Update Menu**, **Process Payment**.
- On a use case diagram, use cases are associated (3) with the actors that perform them.
- Your *system (4)* is
  - whatever you are developing. It might be a small software component, whose actors are just other software components; or it might be a complete application; or it might be a large distributed suite of applications deployed over many computers and devices. Example subsystems are **Meal Ordering Website**, **Meal Delivery Business**, **Website Version 2**.
- A use case diagram can show which use cases are supported by your system or its subsystems.



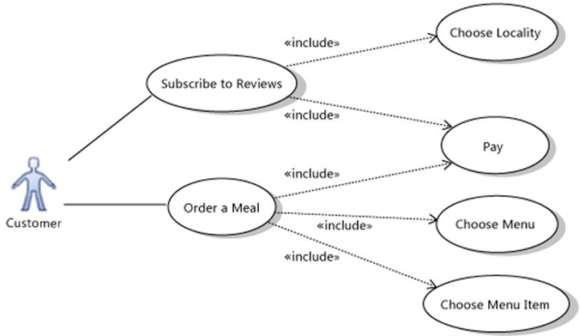
Use Case Modeling Basics (2/5)

- You can draw a **Generalization** link between Actors.
  - The specialized actor, such as Club Customer in the example, inherits the use cases of the generalized actor, such as Customer.
- The association between an actor and a use case can show a *multiplicity* at each end.
  - **1** to state that exactly one instance of this role participates in each link.
  - **1..*** to state that one or more instance of this role participate in each link.
  - **0..1** to state that participation is optional.
  - * to state that zero or more instances of this role participate in the link.
  - In the illustration, one or more restaurants can take part in fulfilling the same meal order.





Use Case Modeling Basics (3/5)



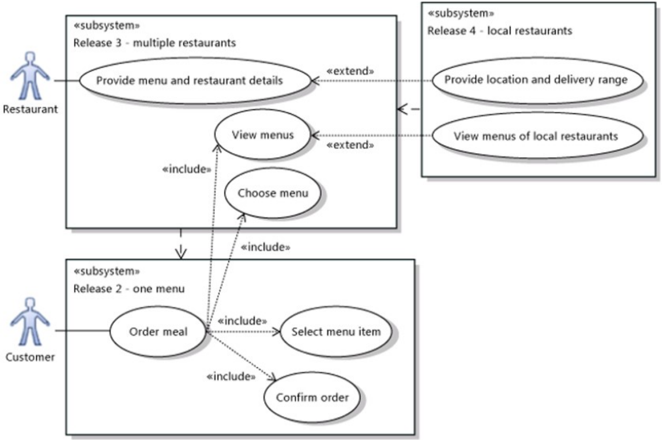
- Use an **Include** relation to show that one use case describes some of the detail of another. In the illustration, **Order a Meal** includes **Pay**, **Choose Menu**, and **Choose Menu Item**. Each of the included, more detailed use cases is a step that the actor or actors might have to perform to achieve the overall goal of the including use case. The arrow should point at the more detailed, included use case.

Use Case Modeling Basics (4/5)

- Use an Extend link to show that one use case may add functionality to another use case under certain circumstances. The arrow should point at the main, extended use case.
- For example, the **Login** use case of a typical Web site can include **Register New User** - but only when the user does not already have an account.

Establishing Requirements

Use Case Modeling Basics (5/5)



- You can use different subsystem boundaries to illustrate different versions of the system.
- Use **Dependency** relations to link subsystems representing different versions or variants.

Establishing Requirements: Software Requirement

Importance of Software Requirements

- Software requirements provide the essential groundwork for software design and development, outlining the functionality and behavior expected from the system.
- A well-written set of requirements will clearly define what a software system must do in order to satisfy customer or user needs.
- Once written, software requirements serve as a way for each stakeholder to do their job effectively and stay in alignment with the other members of the project team.

Establishing Requirements: Software Requirement

Characteristics of Good Software Requirements (1/2)

- Concise: Requirements should be expressed in clear and concise language, avoiding unnecessary complexity and wordiness.
- Precise: Requirements should be specific, well-defined, and unambiguous, leaving no room for interpretation or confusion.
- Clear: Requirements should be free from ambiguity, jargon, and technical terms that may be unclear to stakeholders.
- Complete: Requirements should include all the necessary details and information required for successful implementation, leaving no crucial aspects or functionality undocumented.

Establishing Requirements: Software Requirement

Characteristics of Good Software Requirements (2/2)

- Non-contradictory: Requirements must not contradict each other or any previously established requirements to ensure consistency and coherence.
- Testable: Requirements should be formulated in a way that makes them easily testable and verifiable, allowing for effective validation and quality assurance.
- Prioritized: Requirements should be organized and prioritized based on their importance and impact, ensuring that the most critical aspects are addressed first.

How to write software requirements - The DO's

1. User-Centric Writing:
   - When writing software documentation, adopt the perspective of the user.
   - Focus on their needs, goals, and experiences throughout the document.

BAD REQUIREMENT	GOOD REQUIREMENT
The shopping cart has a list of items and the total purchase price.	As a customer I want to be able to view a list of the items in my cart along with a total purchase price, so that I can decide quickly if I want to move forward with checkout.

How to write software requirements - The DO's

2. Implementation-Neutral Requirements:
   - Formulate requirements without being tied to specific implementation technologies or platforms.
   - Focus on what the software system should achieve rather than how it should be implemented.

BAD REQUIREMENT	GOOD REQUIREMENT
The application must display the onboarding guide using a modal window with a width of 500px and a height of 300px.	The user should be able to view the onboarding guide regardless of what screen they are on.

Establishing Requirements: Software Requirement

How to write software requirements - The DO's

3. Involve stakeholders early and often:
   - Consult stakeholders at the beginning of the requirement process:
     - Get their input on project scope, intended audience, and objectives.
     - Understand their perspectives and expectations to align with project goals.
   - Seek stakeholder input on draft requirements before development:
     - Ensure their needs and preferences are considered.
     - Incorporate valuable insights to refine and enhance the requirements.
   - Notify stakeholders of changes to requirements or product scope during development:
     - Keep stakeholders informed to maintain transparency and manage expectations.
     - Address concerns or potential impact of changes to ensure stakeholder satisfaction.

How to write software requirements - The DO's

4. Analyze, Refine, and Decompose Requirements:
   - Evaluate High-Level Requirements:
     - Assess feasibility, reliability, and verifiability.
     - Resolve issues by adjusting budget, schedule, or modifying requirements.
   - Analyze the Requirements:
     - Ensure completeness, consistency, feasibility, balance, verifiability, and human factors.
   - Define Derived Software Requirements:
     - Complete definition and examine consistency, feasibility, and implementation impact.
     - Monitor size volatility of derived requirements.

How to write software requirements - The DO's

4. Analyze, Refine, and Decompose Requirements:
   - To demonstrate the process, let's break down the requirement on the left into three more specific requirements:

BAD REQUIREMENT	GOOD REQUIREMENT
As a student I need to be able to complete lessons within my eLearning courses.	1. As a student I need to be able to watch video lessons within my eLearning courses. 2. As a student I need to be able to read text-based lessons within my eLearning courses. 3. As a student I need to be able to progress through each lesson in order, without jumping ahead to future lessons within my eLearning courses.

Establishing Requirements: Software Requirement

How to write software requirements - The DO's

5 Specify the Priority of Requirements:

- Identify Decision Makers: those to decide on requirement priorities.
- Establish Priority Labels: a set of priorities such as "must have", "should have", and "nice to have", or a numerical scale from 1 to 10.
- Define Criteria: that justify a requirement's priority level.

PRIORITY	DEFINITION
1	Highest priority, core functionality and/or must have basic customer success in MVP
2	Medium priority, ideally is included in the MVP, as it enables more advanced functionality
3	Lowest priority, would be nice to have, but not required for MVP

- By establishing a prioritization matrix and involving the team in the process, priorities can be set effectively based on the software's end-goals, project budget, and timeline.

Establishing Requirements: Software Requirement

How to write software requirements - The DO's

6. Systematically Track Requirement Changes:
   - Recognize the Need for Change - requirements are subject to changes (unforeseen issues, budget constraints, etc.) during the software development cycle.
   - Implement a Change Management System - establish a structured process to manage requirement changes effectively with roles and responsibilities for stakeholders involved in the change management process.
   - Capture and Document Changes - the reasons, impact, and implications of each change to maintain a comprehensive record.
   - Assess and Analyze Changes - the risks associated with implementing the changes
   - Obtain Approval and Communication - seek approval from stakeholders and communicate requirement changes to project team

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

1. Avoid Ambiguous Language and Technical Jargon:
   - Be Specific and Clear - use precise and unambiguous language to document requirements.
   - Define Key Concepts - define terms such as "easy to use" and "intuitive" to provide a shared understanding as well as specify the functionalities that contribute to these characteristics.
   - Eliminate Technical Jargon -strive for simplicity and clarity to ensure all stakeholders can comprehend the requirements.

BAD REQUIREMENT	GOOD REQUIREMENT
The system should provide an intuitive user interface that is easy to use.	The user should always be able to access the Homepage in one-click.

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

2. Avoid Excluding Acceptance Criteria:
   - Acceptance criteria are the conditions a software product must meet to be accepted by a user, a customer, or other systems.
   - Clear acceptance criteria are essential for successful software development.
   - They prevent misinterpretation and ensure alignment with stakeholders.
   - Acceptance criteria provide measurable benchmarks for quality assurance.
   - Including acceptance criteria enhances communication and mitigates misunderstandings.
   - They enable validation of requirement fulfillment and satisfaction of stakeholder expectations.

How to write software requirements - The DONT's

2. Avoid Excluding Acceptance Criteria:

BAD REQUIREMENT	GOOD REQUIREMENT
The user must be able to search for employees by department.	The user must be able to search for employees by department. 1. The user must be able to select a department from a list of available departments. 2. The search must return a list of employees in the selected department. 3. The search results must include the employee's name, department, and title. 4. The search must be completed within 5 seconds.

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

3 Avoid Neglecting to use a Standardized Format:

- Standardized formats for software requirements enhance clarity and understanding.
- Specification consistency improves communication among stakeholders.
- A standardized format facilitates future maintenance and updates.
- Using a common format reduces ambiguity and misinterpretation.

BAD REQUIREMENTS	GOOD REQUIREMENTS
1. The application must be able to track and display sales data in real-time. 2. Users must be able to access the application from any device with a web browser. 3. All pages must be secure and password protected.	Admins must be able to pull a real-time sales data report. Users must be able to access the application from any device with a web browser. Users should be required to enter a password before accessing any pages of the application.

How to write software requirements - The DONT's

4. Avoid Making Technical Assumptions:
   - Avoid making technical assumptions in software requirements.
   - Clearly specify technical requirements in nonfunctional requirements.
   - Focus on functionality from the user's perspective rather than specific technologies.
   - Improve clarity and flexibility by decoupling functional requirements from technical implementation details.

BAD REQUIREMENT	GOOD REQUIREMENT
Real-time analytics of user activity data should be stored in the relational database.	Admins should have access to real-time analytics of user activity data.

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

5. Avoid Adding Conflicting Requirements:
   - Conflicting requirements can cause confusion, ambiguity, and project delays.
   - Carefully review requirements to identify and resolve conflicts.
   - Revise conflicting requirements to provide clarity and consistency.
   - Ensure stakeholders and development teams are aware of and address conflicting requirements.

BAD REQUIREMENT	GOOD REQUIREMENT
1. Team leaders should have full access to reports. 2. Admins should be able to assign "read only" or "edit" permissions to each team leader individually.	1. Team leaders should, by default, be given "edit" access to reports. 2. Admins should be able to change the reports permissions for individual team leaders to assign them either "edit" or "read only" access.

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

6. Avoid Neglecting Non-Functional Requirements:
   - Non-functional requirements are crucial in software requirement specifications.
   - Neglecting non-functional requirements can result in systems that fail to meet stakeholder expectations.
   - Non-functional requirements define system quality, security, and reliability.
   - Include non-functional requirements in the software requirement specification document.
   - Address areas such as performance, scalability, security, usability, and maintainability.

Establishing Requirements: Software Requirement

How to write software requirements - The DONT's

6. Avoid Neglecting Non-Functional Requirements:
   - Nonfunctional requirements should address:

Security	Will you be storing PII or other sensitive data? Does your company have any specific security standards that should be adhered to?
Capacity	What kind of data storage requirements do you have? Do you expect your needs to change over time?
Compatibility	What are the minimum hardware requirements? What are the technical limitations that should be considered? Are there any specific external interface requirements?
Reliability	Do users need 24/7 access? What are the acceptable down times for your system to still meet a user's basic needs?
Scalability	What is the maximum load that should be able to be handled? Consider both data and user traffic.
Usability	Are there specific UX standards that should be followed?

Establishing Requirements: Software Requirement

Group Assignment

List of members for each group?

Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: https://vincentnofong.com/
- Uni website: https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Rules

1. Feel free to ask questions in class, unless they are too "personal".
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Three: Prototyping

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Prototyping

What is a Prototype?

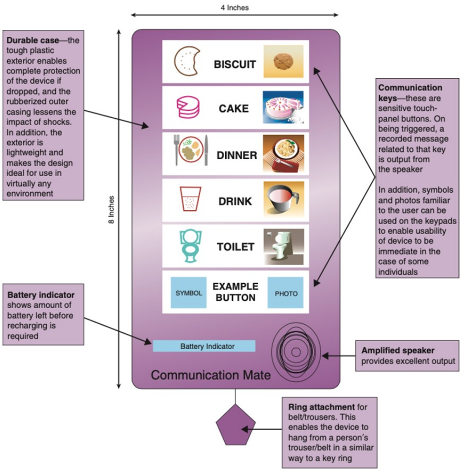
- A prototype is a representation of a design that enables stakeholders to interact with it.
- In various design fields, prototypes can take the form of small-scale models.
- Examples of prototypes include miniature cars, miniature buildings, or miniature towns.
- Prototypes serve as tangible representations that help visualize and evaluate the design concept.
- They allow stakeholders to provide feedback, test functionality, and make informed decisions before final implementation.

Prototyping

What is a Prototype in Interaction Design?

- In interaction design, a prototype serves various purposes and can take different forms.
  - Examples include series of screen sketches, storyboards, PowerPoint slide shows, video simulations, physical prototypes, loosely connected electronic elements, animations of product use, and software with limited functionality.
- These prototypes help visualize and demonstrate the intended user experience and functionality of the design.
- They allow for user feedback, usability testing, and iterative design improvements before final implementation.
- Choosing the appropriate prototype format depends on the specific project requirements, goals, and resources available.

Example Interaction Design Prototype: A paper-based prototype of a handheld device to support an autistic child



Prototyping

Why Prototype?

- It facilitates evaluation and feedback in interaction design.
- Prototypes provide a tangible and interactive representation of the design, making it easier for stakeholders to comprehend and engage with.
- Prototypes allow ideas to be tested and explored, helping designers make informed decisions.
- Reflection is encouraged through the prototyping process, promoting critical assessment and refinement of designs.
- Prototypes help answer questions and support designers in choosing between alternative design solutions.

Low-Fidelity Prototyping

- Low-fidelity prototyping involves using a medium that is different from the final product, such as paper or cardboard.
- It is a simple, quick, and cost-effective method that allows for easy modifications.
- Examples of low-fidelity prototyping include:
  - sketches of screens and task sequences,
  - index cards or sticky notes,
  - storyboards, and,
  - the Wizard-of-Oz technique

Low-Fidelity Prototyping: Storyboards

- Storyboards are a series of sketches that depict the progression of a user through a task using the product.
- They are frequently used in conjunction with scenarios to provide more detail and enable role-playing.
- Storyboards offer a visual representation of the user experience and help in understanding the flow of interaction with the product.

Prototyping

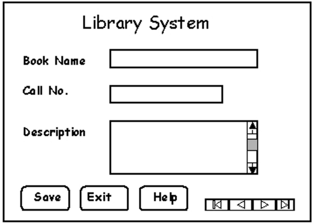
Low-Fidelity Prototyping: Storyboards -An Example

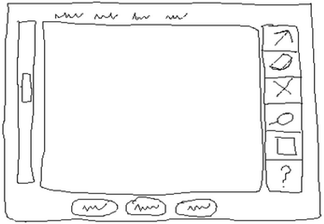
Low-Fidelity Prototyping: Sketching

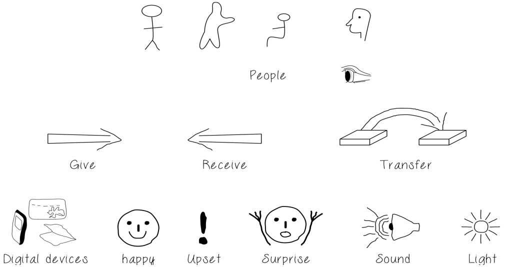
- Sketching is a common technique used in low-fidelity prototyping.
- It allows for quick and easy visualization of design ideas and concepts.
- Drawing ability should not be a limitation, as simple symbols and shapes can effectively communicate design concepts.
- Sketching helps to explore and iterate on design ideas before investing significant time and resources in high-fidelity prototypes.

Prototyping

Low-Fidelity Prototyping: Sketching - Examples





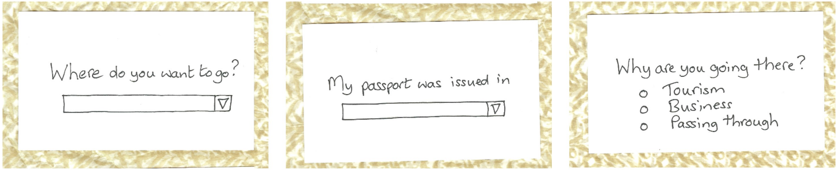


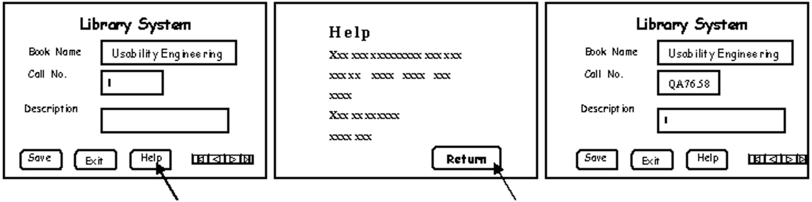
Low-Fidelity Prototyping: Index Cards

- Index cards (3 x 5 inches) can be used for prototyping.
- Each card represents a specific element of interaction in the design.
- During evaluation, the cards can be stepped through to simulate the flow and interaction of the system.
- Index cards provide a tangible and flexible way to prototype and iterate on design ideas.

Prototyping

Low-Fidelity Prototyping: Index Cards - Examples





Don't know how to input → **Help screen for *Call No.* Field** → **Finish input *Call No.***

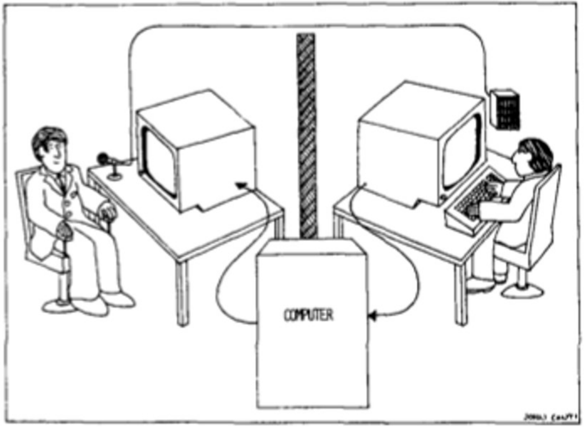
Click Help → **Click *Return* after reading** → **Ready to input the next field**

Low-Fidelity Prototyping: Wizard-of-Oz

- Wizard-of-Oz prototyping involves simulating an interactive system where a human responds to the participant's input instead of using an actual functioning system.
- Participants believe they are interacting with a computer, unaware that a human is behind the scenes.
- This technique is commonly used in the early stages of design to gain insights into people's expectations and behaviors.
- Wizard-of-Oz prototyping allows designers to observe user interactions and collect valuable feedback before investing in the development of a fully functional system.

Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz



Example: Aardvark startup - acquired by Google and subsequently abandoned/shutdown by Google

Low-Fidelity Prototyping: Wizard-of-Oz

Some of the potential ethical issues associated with this approach include (1/2):

- Deception: Misleading participants about automated system interaction raises ethical concerns and impacts trust.
- Informed Consent: Participants must be fully informed about the prototyping method, including the human operator, and provide consent.
- Privacy and Data Protection: Protecting participant information and ensuring confidentiality is essential. Designers must handle sensitive data responsibly and adhere to privacy regulations.

Prototyping

Low-Fidelity Prototyping: Wizard-of-Oz

Some of the potential ethical issues associated with this approach include (2/2):

- Emotional Well-being: Participants may have emotional reactions upon discovering they interacted with a human instead of a real system. Designers should monitor and support participants to address any negative emotions.
- Psychological Impact: Consider potential psychological impacts and ensure participant well-being during Wizard-of-Oz prototyping.

High-Fidelity Prototyping

- Utilizes materials resembling the final product
- Resembles the final system more closely than low-fidelity prototypes
- Involves integration of existing hardware and software components
- Caution: Participants may perceive it as a complete system, leading to potential compromises.

Low-Fidelity vs High-Fidelity Prototyping

Type	Advantages	Disadvantages
Low-fidelity prototype	Lower development cost Evaluates multiple design concepts Useful communication device Addresses screen layout issues Useful for identifying market requirements Proof of concept	Limited error checking Poor detailed specification to code to Facilitator-driven Limited utility after requirements established Limited usefulness for usability tests Navigational and flow limitations
High-fidelity prototype	Complete functionality Fully interactive User-driven Clearly defines navigational scheme Use for exploration and test Look and feel of final product Serves as a living specification Marketing and sales tool	More resource-intensive to develop Time-consuming to create Inefficient for proof-of-concept designs Not effective for requirements gathering

Compromises in Prototyping

- Prototyping entails compromises in various aspects.
- Software-based prototypes may have compromises such as slow response, sketchy icons, or limited functionality.
- In-the-wild prototypes may be operational but not fully robust.
- Two common types of compromise are horizontal, offering a wide range of functions with little detail, and vertical, providing detailed functionality for only a few functions.
- Another common compromise is between robustness and changeability.
- Caution is advised as prototypes should not be mistaken for the final engineered product, considering the compromises made.

Concrete Design: Enhancing User Experience (1/2)

- By focusing on concrete design, designers can enhance the user experience by refining visual elements, considering user characteristics and context, ensuring accessibility, accommodating localization needs, and incorporating diverse cultural perspectives.
- This approach leads to more inclusive, engaging, and culturally sensitive designs.

Prototyping

Concrete Design: Enhancing User Experience (2/2)

- Concrete design focuses on refining and specifying design elements - color, icons, buttons, and interaction devices.
- Design for inclusiveness by accommodating diverse user characteristics - consider input and output modes that suit user preferences and capabilities.
- Use Web Content Accessibility Guidelines to ensure inclusivity.
- Tailor the design to support different languages and cultural contexts -consider adapting navigation, icons, and metaphors for global users.
- Integrate indigenous knowledge into the design process - how design elements can reflect diverse cultural perspectives.

Generating Prototypes

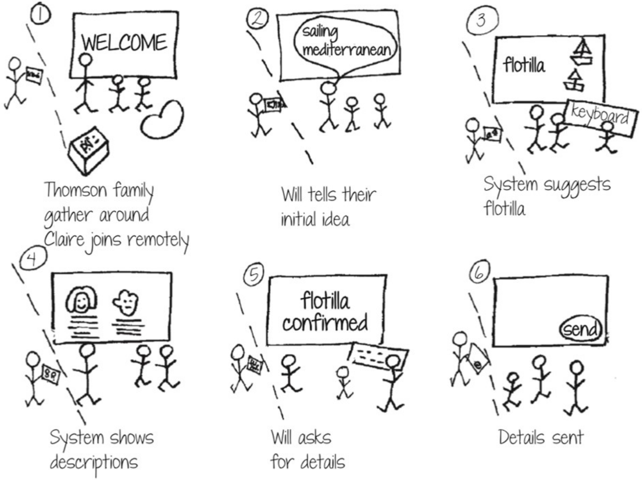
- By generating prototypes, designers can gain insights into user interactions, identify design considerations, and foster a collaborative design process.
- Prototyping serves as a valuable tool in the iterative design cycle, aiding in creating user-centered and impactful solutions.
- Benefits of Prototyping:
    - Visualizes the user's journey and interaction with the product.
    - Promotes early exploration of design issues and considerations.
    - Facilitates effective communication and collaboration among team members.
    - Enables iterative design and feedback to refine the user experience.

Generating Prototypes: From Storyboards to Card-Based Prototypes

- Storyboard Generation:
  - Break down scenarios into sequential steps.
  - Create a scene for each step to visualize the user's interaction.
  - Storyboarding prompts designers to consider design issues effectively.
- Card-Based Prototype Generation:
  - Utilize storyboards or use cases as a basis for creating a card-based prototype.
  - For each step in the use case, identify the required interaction elements.
  - Draw a card that captures the specific interaction element for that step.

Prototyping

Generating Storyboards



Prototyping

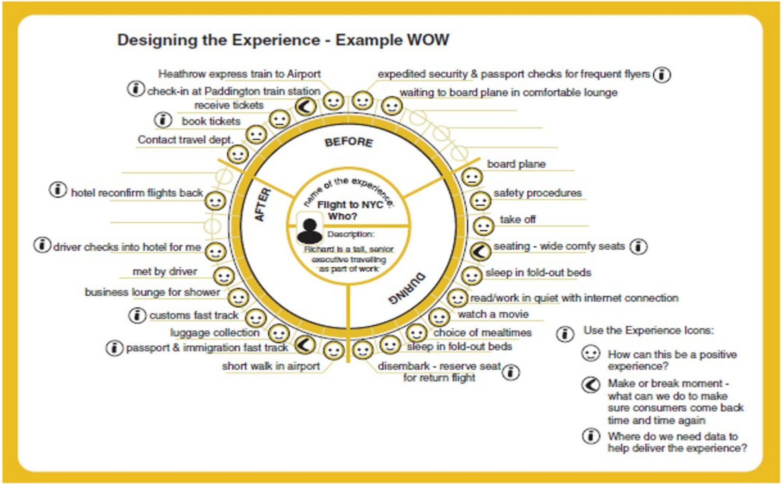
Generating Card-Based Prototypes

Mapping the Overall Experience - Not examinable

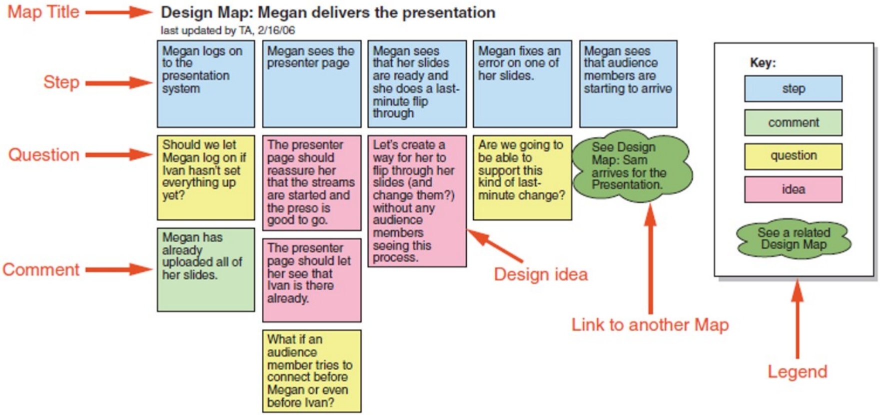
- By utilizing design maps, customer journey maps, experience maps, and user flows, designers can gain valuable insights into the overall user experience.
- These visual representations aid in identifying key touchpoints, understanding user needs, and informing design decisions to create meaningful and engaging experiences.
- Benefits of Mapping the Overall Experience:
  - Provides a holistic view of the user's journey and interactions.
  - Identifies pain points, opportunities, and areas for improvement.
  - Enhances empathy and understanding of the user's perspective.
  - Facilitates collaboration and communication among stakeholders.

Prototyping

Mapping the Overall Experience: An Experience Map as a Wheel - Not examinable



Mapping the Overall Experience: An Experience Map as a Timeline- Not examinable



Prototyping

Construction: Physical Computing

- By utilizing toolkits such as Arduino, LilyPad, Raspberry Pi, BBC micro:bit, and MaKey MaKey, designers can bring their ideas to life through physical computing.
- These platforms provide a foundation for building interactive prototypes, fostering creativity, and enabling innovative user experiences.
- Benefits of Physical Computing:
  - Enables the integration of physical components and interactivity into prototypes.
  - Facilitates the exploration of tangible and interactive user experiences.
  - Encourages interdisciplinary collaboration between designers, developers, and makers.
  - Supports rapid prototyping and iterative design processes.

Prototyping

The Four Major Software Prototyping Techniques

- When it comes to software prototyping, there are four prominent methods to consider.
- These techniques can greatly simplify the process of creating improved and presentable software prototypes.
  1. Incremental Prototyping
  2. Throwaway Prototyping
  3. Extreme Prototyping
  4. Evolutionary Prototyping
- While there are other methods available, these four techniques stand out as the most effective options for creating enhanced and presentable software prototypes.
- Each method offers a slightly different approach, but all contribute to the overall goal of building superior software

Prototyping

The Four Major Software Prototyping Techniques - Incremental Prototyping: (1/2)

- Incremental prototyping is a software development approach that involves creating iterative versions of a prototype, with each version being an improvement over the previous one.
- This method follows a linear development process, where features are added, design is enhanced, and bugs are fixed at each stage.
- By following incremental prototyping, you can gradually enhance your software prototype and ensure a smooth development process.
- Each iteration brings you closer to a more refined and reliable end product.

The Four Major Software Prototyping Techniques - Incremental Prototyping (2/2)

Benefits of Incremental Prototyping:

- Easy and quick creation of software prototypes.
- Allows for frequent testing and validation within a short timeframe.
- Ensures continuous improvement and refinement of the prototype.
- Minimizes errors and reduces the risk of failure in the final software.

Note: Incremental Prototyping focuses on Building Better Software Prototypes

The Four Major Software Prototyping Techniques - Throwaway (Rapid) Prototyping (1/2)

- Throwaway prototyping, also known as close-ended prototyping, is a method where the main goal is to refine a specific aspect of the prototype.
- It is not intended for integration into the final software product - used to gather user feedback and make improvements based on that feedback.
- With throwaway prototyping, you can efficiently collect valuable insights from users and make informed decisions about the next steps in the development process.
- By focusing on specific aspects, you can create a more effective and user-centered prototype.

The Four Major Software Prototyping Techniques - Throwaway (Rapid) Prototyping (2/2)

Advantages of Throwaway Prototyping

- Allows for targeted refinement of specific prototype features.
- Feedback-driven approach for iterative improvements.
- Reusable components can be retained for future use.
- Saves time and effort by eliminating the need for extensive documentation.

Note: Throwaway Prototyping Focuses on Specific Aspects

The Four Major Software Prototyping Techniques - Extreme Prototyping (1/2)

- Extreme prototyping is a valuable approach primarily used for developing web applications, consisting of three stages:
  - Building a Static Prototype: Creating HTML pages that represent the user interface and overall structure of the application.
  - Programming Screens with a Stimulated Service Layer: Adding functionality to the screens by simulating the service layer that interacts with the backend systems.
  - Implementing the Services: Integrating the actual services and backend functionality into the prototype.
- By adopting extreme prototyping, developers can mitigate risks, optimize performance, and create web applications that effectively meet user needs and expectations.

The Four Major Software Prototyping Techniques - Extreme Prototyping (2/2)

Benefits of Extreme Prototyping:

- Early Bug Detection: Identifying and addressing bugs and issues before the application reaches the production stage, resulting in cost savings and improved efficiency.
- Enhanced User Engagement: Prioritizing user experience and performance at each stage to meet user expectations and increase engagement.
- Iterative Development: Iteratively refining the prototype to ensure it evolves into a final product that effectively engages users.

The Four Major Software Prototyping Techniques - Evolutionary Prototyping (1/2)

- Evolutionary prototyping, also known as breadboard prototyping, is a fundamental and developer-focused approach to prototyping.
- This method involves creating a robust prototype that incorporates a comprehensive set of features and functionalities based on well-understood requirements.
- As the process evolves, additional enhancements and improvements are integrated, resulting in a refined prototype.
- With evolutionary prototyping, developers can build robust prototypes that evolve alongside the project, resulting in a refined and effective end product.

The Four Major Software Prototyping Techniques - Evolutionary Prototyping (2/2)

Benefits of Evolutionary Prototyping:

- Improved Understanding: Developers gain understanding of the project requirements and refine the prototype accordingly.
- Iterative Refinement: The iterative nature of this method allows for ongoing improvements and adjustments to align the prototype with user expectations.
- Risk Mitigation: Early identification of potential issues enables proactive problem-solving and risk mitigation.

Note: Evolutionary Prototyping Focuses on Building Robust and Refined Prototypes

Prototyping

Reading Assignment: The Windws$^{(R)}$ 95 User Interface Prototyping and Usability Testing

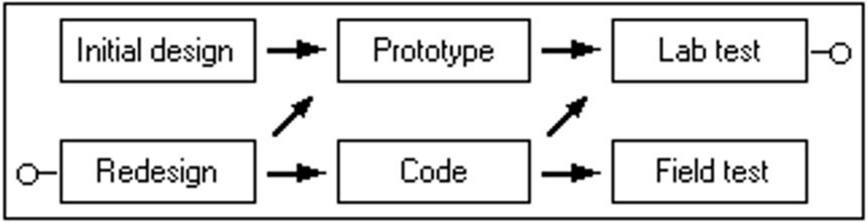


Figure 1: Windows 95 Iterative Design Process

Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: `https://vincentnofong.com/`
- Uni website: `https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385`
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Introduction

Rules

1. Feel free to ask questions in class, unless they are too "personal".
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Four: Data Gathering and Analysis

Course Instructor: Vincent M. Nofong, Ph.D.

July 9, 2024

Capturing Data: Methods and Considerations

- Multiple Data Capture Options: Notes, audio, video, and photographs can be used individually or combined.
- Consider the nature of the information to be captured and the research goals.
- Determine which method(s) best align with the research objectives and offer the most comprehensive data.
- Challenges and Advantages:
  - Notes: Ideal for capturing textual info, but lack visual context.
  - Photographs: Provide visual representation, but limited in capturing dynamic interactions.
  - Audio: Ideal for capturing spoken conversations and verbal nuances.
  - Video: Rich visual and auditory info, but requires more storage & processing.

Interviews

- Interviews are a valuable research method for gathering data.
- **Unstructured** interviews offer flexibility and richness but lack replicability.
- **Structured** interviews provide replicable results but may sacrifice richness.
- **Semi-structured** interviews strike a balance between richness and replicability.
- **Focus groups** involve group interviews to encourage interactive discussions and diverse perspectives.
- Select the interview method based on the research goals and the desired level of structure.

Interview Questions

- Interview questions can be categorized as **closed** or **open-ended**.
- **Closed questions** have a predetermined answer format, while **open questions** allow for more diverse responses.
- Closed questions are easier to analyze, but open questions provide richer insights.
- When formulating interview questions, avoid:
  - Long questions
  - Compound sentences — split them into two
  - Jargon and language that the interviewee may not understand
  - Leading questions that make assumptions, E.g., “why do you like . . . ?”
  - Unconscious biases, for instance, gender stereotypes

Data Gathering and Analysis

Running the interview

- **Introduction**: Introduce yourself, state the interview goals, address ethical considerations, request recording, and present the informed consent form.
- **Warm-up**: Begin with easy and non-threatening questions to establish rapport.
- **Main body**: Present questions in a logical order, addressing the desired topics.
- **Cooling-off period**: Include a few easy questions towards the end to defuse tension.
- **Closure**: Thank the interviewee, signal the end of the interview (e.g., switching off the recorder).

Doing Interviews Remotely

- Remote interviews and focus groups using digital conferencing systems like Teams and Zoom, along with collaboration platforms such as Miro, are commonly used.
- Advantages:
  - Participants are in their own environment and feel more relaxed.
  - Participants don't have to travel, saving time and effort.
  - Participants don't need to worry about their attire.
  - Remote interviews offer increased anonymity for sensitive issues.
  - Participants have the freedom to leave the interview at any time.

Enriching The Interview Process

- Incorporating props in the interview process, such as personas, prototypes, and scenarios, enhances engagement and deepens insights.
- These tools provide a tangible and relatable context for discussions, helping to elicit valuable information from the interviewees' perspectives.

Questionnaires

- Questionnaires are a valuable data collection method that uses closed and open-ended questions.
- Closed questions are easier to analyze and can be processed using computer software.
- Online dissemination of questionnaires allows for reaching large populations efficiently.
- However, sampling can be challenging when the size of the target population is unknown, as is often the case in online evaluations.

Questionnaire Design Tips

- **Question Order**: Be mindful of the impact of question order, as it can influence participant responses.
- **Tailored Versions**: Create different versions of the questionnaire for different target populations, ensuring relevance and engagement.
- **Clear Instructions**: Provide clear and concise instructions on how to complete the questionnaire to minimize confusion.
- **Length Considerations**: Assess the length of the questionnaire and consider if it may be too long.
- **Layout and Pacing**: Pay attention to the layout and pacing of the questionnaire to maintain participant interest and ensure a smooth completion experience.

Question and Response Formats

- Closed-ended Responses:
  - **Radio Buttons**: Choose a single response from a predefined list.
  - **Check Boxes**: Select multiple responses from a predefined list.
  - **Rating Scales**: Assign ratings on a numerical or descriptive scale.
  - **Likert Scales**: Measure agreement or disagreement on a statement using a range of response options.
  - **Semantic Differential Scales**: Rate concepts based on opposing adjectives or dimensions using a scale.
- Open-ended Responses:
  - Allow participants to provide unrestricted and personalized answers, without predefined options.

Encouraging a Good Response (1/2)

- **Clarify the Purpose**: Clearly communicate the objective of the study to participants, ensuring they understand its importance and relevance.
- **Ensure Anonymity**: Assure participants that their responses will remain anonymous, fostering a sense of confidentiality and encouraging honest feedback.
- **Careful Questionnaire Design**: Develop the questionnaire thoughtfully, considering the clarity and flow of questions.
- **Short Version Option**: Offer a condensed version of the questionnaire for participants with limited time, increasing the likelihood of their participation.

Encouraging a Good Response (2/2)

- **Prompting Messages**: Send follow-up reminders or prompts to participants who have not yet completed the questionnaire, gently encouraging their response.
- **Incentives**: Consider providing incentives, such as vouchers or small rewards, to motivate participants and show appreciation for their participation.
- **Response Rate Expectations**: While a response rate of around 40% is generally acceptable, lower response rates are common in practice and should be taken into account when analyzing the data.

Administering Questionnaires (1/2)

- **Timeline Planning**: Establish a clear timeline for questionnaire administration, considering the start and end dates, as well as any intermediate milestones.
- **Design Offline**: Begin the questionnaire design process offline, outlining the structure, content, and response formats before moving to an online platform.
- **Program/Complete Online Template**: Transfer the questionnaire design to an online survey tool or platform, ensuring all questions, response options, and skip logic are accurately implemented.

Administering Questionnaires (2/2)

- **Test the Survey**: Conduct thorough testing of the survey to ensure it functions as intended. Verify that question branching, skip patterns, and response validation work correctly.
- **Test with a Group**: Validate the survey's clarity and effectiveness by piloting it with a group of individuals who are not part of the actual survey population. Seek their feedback and make necessary improvements.
- **Participant Recruitment**: Develop a strategy for participant recruitment, whether through random sampling, targeted invitations, or other appropriate methods. Ensure the chosen sample is representative of the desired population.

Observation: Direct Observation in the Wild

Conducting observations of users in their natural environments.

- Structuring Frameworks: Utilizing frameworks or protocols to guide the observation process.
- Degree of Participation: Observing users passively or actively participating in their activities.
- Ethnography: Employing ethnographic methods to gain deep insights into users' behaviors, cultures, and contexts.

Data Gathering and Analysis

Observation: Direct Observation in a Controlled Environments

Conducting observations in controlled settings, such as usability labs or simulated environments.

- Think-Aloud Technique: Prompting users to verbalize their thoughts and actions during the observation.

Data Gathering and Analysis

Observation: Indirect Observation

Tracking users' activities without direct physical presence.

- Diaries: Requesting participants to keep diaries or journals documenting their experiences or activities.
- Interaction Logging, Web Analytics, and Data Scraping: Collecting data on user interactions and behaviors through digital platforms.
- Remote Collection of Video and Photographs: Utilizing remote technologies like drones to capture visual data.
- Wearable Sensors and Social Media: Leveraging wearable devices or monitoring users' activities through social media platforms.

Adapting the techniques for different participants

Visual representation of a Likert scale for children

GPS tracker on a cat

Quantitative and Qualitative Data Analysis

- Quantitative Data
  - Expressed as numbers.
  - Focuses on size, magnitude, or amount.
  - Allows for numerical analysis.
- Qualitative Data
  - In the form of words and images.
  - Captures the nature of elements.
  - Analyzed through themes, patterns, or stories.

Quantitative and Qualitative Data Analysis

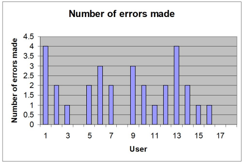
- Quantitative Analysis
  - Utilizes numerical methods.
  - Aims to measure and quantify data.
  - Focuses on numerical relationships.
- Qualitative Analysis
  - Explores the meaning and context.
  - Identifies themes and patterns.
  - Emphasizes the narrative and descriptive aspects.
- Data Manipulation
  - Exercise caution when manipulating data and numbers.
  - Consider the appropriate treatment based on the data type and research objectives.

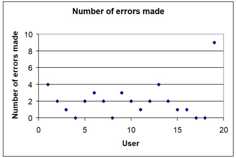
Basic Quantitative Analysis (1/2)

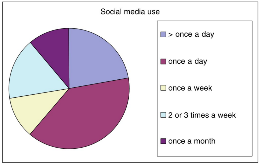
- Averages:
  - Mean: Sum of values divided by the number of data points.
  - Median: Middle value of the data when ranked.
  - Mode: Value that appears most frequently in the data.
- Percentages:
  - Expresses a proportion or relative amount.
  - Useful for comparing parts to the whole.
- Cautions in Data Interpretation:
  - Exercise caution to avoid misleading interpretations.
  - Consider the context and potential biases in the data.

Basic Quantitative Analysis (2/2)

- Graphical Representations:
  - Provide an overview of the data.
  - Enhance visual understanding and comparison.







- Remember:
  - Choose appropriate analysis techniques for accurate interpretation and meaningful insights.
  - Use graphs and visualizations effectively to communicate data patterns.

Basic Qualitative Analysis

- Recurring Patterns or Themes
  - Identify recurring patterns or themes within the data.
- Categorizing Data:
  - Categorization involves organizing data into meaningful groups.
  - The categorization scheme can be emergent, developing as analysis progresses, or pre-specified based on prior knowledge or research objectives.
- Critical Incidents:
  - Focus on critical incidents that highlight key events or moments of significance.
  - By examining critical incidents, researchers gain valuable insights and understanding.

Data Gathering and Analysis

Presenting Findings

- **Support Claims with Data**: Ensure your claims are backed by solid evidence from your data analysis.
- **Tailor Presentation to Audience**: Consider the audience, purpose, and research methodology when deciding how to present findings.
- **Graphical Representations**: Utilize graphs and visualizations to effectively convey information.
- Additional Techniques:
  - **Storytelling**: Use narratives to create engaging and relatable scenarios.
  - **Summarization**: Provide concise summaries of the research findings.

Tools for Data Analysis

- Spreadsheets: User-friendly with basic graphing capabilities.
- Statistical packages (e.g., SPSS): Designed for quantitative analysis.
- Qualitative data analysis tools:
  - Categorization and theme-based analysis.
  - Quantitative analysis of text-based data.
- Notable tools:
  - NVivo (http://www.qsrinternational.com/)
  - Atlas.ti (http://atlasti.com/)
  - QDA Miner (https://provalisresearch.com/)
  - Orange (https://orangedatamining.com/)

Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

July 11, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: https://vincentnofong.com/
- Uni website: https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Introduction

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Introduction

Rules

1. Feel free to ask questions in class, unless they are too "personal".
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Five: Cognitive Aspects of Design

Course Instructor: Vincent M. Nofong, Ph.D.

July 11, 2024

What is cognition?

Cognitive Aspects of Design

What is cognition?

- Cognition encompasses various mental processes such as **thinking, remembering, learning, decision-making, and more**.
- Different ways to classify cognition include:
  - Experiential vs. reflective cognition (Norman, 1993): Distinguishes between immediate, intuitive experiences and deliberate, analytical thinking.
  - Fast vs. slow thinking (Kahneman, 2011): Describes the distinction between quick, instinctive thinking and deliberate, effortful thinking.

Cognitive Aspects of Design

Why understand users?

- Interacting with technology involves cognitive processes.
- It is crucial to consider the cognitive abilities and limitations of users.
- Understanding users helps:
  - Determine users' capabilities and limitations.
  - Identify and explain user problems.
  - Provide theories, models, guidance, and methods for designing better interactive products.

Cognitive Aspects of Design

Cognitive Processes

- Attention: The ability to selectively focus on relevant information.
- Perception: The process of interpreting sensory information and giving it meaning.
- Memory: The encoding, storage, and retrieval of information.
- Learning: Acquiring knowledge or skills through experience or instruction.
- Reading, Speaking, and Listening: Language-related processes involved in communication.
- Problem-Solving, Planning, Reasoning, and Decision-Making: Higher-level cognitive abilities used to solve problems, make plans, think logically, and make informed choices.

Cognitive Aspects of Design

Attention

- Selecting things to concentrate on from the surrounding stimuli.
- Enables us to focus on relevant information.
- Involves both audio and visual senses.
- Focused and divided attention allow us to be selective but limit our ability to track all events.
- Structuring information at the interface to capture users' attention:
  - Use perceptual boundaries such as windows.
  - Incorporate **color**, video, sound, and flashing lights.

Attention - Activity 1: Find the price for a double room at the Quality Inn in Bedford

Pennsylvania
Bedford Motel/Hotel: Crinaline Courts
(814) 623-9511 S: $118 D: $120
Bedford Motel/Hotel: Holiday Inn
(814) 623-9006 S: $129 D: $136
Bedford Motel/Hotel: Midway
(814) 623-8107 S: $121 D: $126
Bedford Motel/Hotel: Penn Manor
(814) 623-8177 S: $119 D: $125
Bedford Motel/Hotel: Quality Inn
(814) 623-5189 S: $123 D: $128
Bedford Motel/Hotel: Terrace
(814) 623-5111 S: $122 D: $124
Bradley Motel/Hotel: De Soto
(814) 362-3567 S: $120 D: $124
Bradley Motel/Hotel: Holiday House
(814) 362-4511 S: $122 D: $125
Bradley Motel/Hotel: Holiday Inn
(814) 362-4501 S: $132 D: $140
Breezewood Motel/Hotel: Best Western Plaza
(814) 735-4352 S: $120 D: $127
Breezewood Motel/Hotel: Motel 70
(814) 735-4385 S: $116 D: $118

Attention - Activity 2: Find the price of a double room at the Holiday Inn in Columbia

South Carolina

City	Motel/Hotel	Area code	Phone	Rates Single	Rates Double
Charleston	Best Western	803	747-0961	$126	$130
Charleston	Days Inn	803	881-1000	$118	$124
Charleston	Holiday Inn N	803	744-1621	$136	$146
Charleston	Holiday Inn SW	803	556-7100	$133	$147
Charleston	Howard Johnsons	803	524-4148	$131	$136
Charleston	Ramada Inn	803	774-8281	$133	$140
Charleston	Sheraton Inn	803	744-2401	$134	$142
Columbia	Best Western	803	796-9400	$129	$134
Columbia	Carolina Inn	803	799-8200	$142	$148
Columbia	Days Inn	803	736-0000	$123	$127
Columbia	Holiday Inn NW	803	794-9440	$132	$139
Columbia	Howard Johnsons	803	772-7200	$125	$127
Columbia	Quality Inn	803	772-0270	$134	$141
Columbia	Ramada Inn	803	796-2700	$136	$144
Columbia	Vagabond Inn	803	796-6240	$127	$130

Cognitive Aspects of Design

Findings Based on Activity 1 and 2

- Tullis (1987) found that the two screens produced quite different results
  - Activity 1: Took an average of 5.5 seconds to search
  - Activity 2: Took 3.2 seconds to search
- Why, since both displays have the same density of information (31percent)?

Cognitive Aspects of Design

Findings Based on Activity 1 and 2

- Tullis (1987) found that the two screens produced quite different results
  - Activity 1: Took an average of 5.5 seconds to search
  - Activity 2: Took 3.2 seconds to search
- Why, since both displays have the same density of information (31percent)?
- Spacing:
  - In Activity 1, the information is bunched up together, making it hard to search
  - In Activity 2, the characters are grouped into vertical categories of information making it easier

Designing for Attention

Designing for Attention

- Make important information salient and noticeable.
- Use techniques such as color, ordering, spacing, underlining, sequencing, and animation to make elements stand out.
- Avoid cluttering the interface with excessive information.
- Keep interfaces simple and clean for improved usability.
- Consider designing interfaces that support effective switching between tasks and returning to previous contexts.

Perception

- Perception involves acquiring information from the environment and translating it into meaningful experiences.
- Design representations that are easily perceivable by users.
- Ensure that text is legible and easily readable.
- Create icons that are distinct and recognizable.
- Consider visual clarity and use appropriate visual cues to enhance perception.

Cognitive Aspects of Design

Perception - Activity 3: Is color contrast good? Find “Italian”

Black Hills Forest	Peters Landing	Jefferson Farms	Devlin Hall
Cheyenne River	Public Health	Psychophysics	Positions
Social Science	San Bernardino	Political Science	Hubard Hall
South San Jose	Moreno Valley	Game Schedule	Fernadino Beach
Badlands Park	Altamonte Springs	South Addision	Council Bluffs
Juvenile Justice	Peach Tree City	Cherry Hills Village	Classical Lit
Results and Stats	Highland Park	Creative Writing	Sociology
Thousand Oaks	Manchesney Park	Lake Havasu City	Greek
Promotions	Vallecito Mts.	Engineering Bldg	Wallace Hall
North Palermo	Rock Falls	Sports Studies	Concert Tickets
Credit Union	Freeport	Lakewood Village	Public Radio FM
Wilner Hall	Slaughter Beach	Rock Island	Children's Museum
Performing Arts	Rocky Mountains	Deerfield Beach	Writing Center
Italian	Latin	Arlington Hill	Theater Auditions
Coaches	Pleasant Hills	Preview Game	Delaware City
McKees Rocks	Observatory	Richland Hills	Scholarships
Glenwood Springs	Public Affairs	Experts Guide	Hendricksville
Urban Affairs	Heskett Center	Neff Hall	Knights Landing
McLeansboro	Brunswick	Grand Wash Cliffs	Modern Literature
Experimental Links	East Millinocket	Indian Well Valley	Studio Arts
Graduation	Women's Studies	Online Courses	Hughes Complex
Emory Lindquist	Vacant	Lindquist Hall	Cumberland Flats
Clinton Hall	News Theatre	Fisk Hall	Central Village
San Luis Obispo	Candlewood Isle	Los Padres Forest	Hoffman Estates

Cognitive Aspects of Design

Perception - Activity 4: Are borders & white space better?

Find “French”

Webmaster Russian Athletics Go Shockers Degree Options Newsletter	Curriculum Emergency (EMS) Statistics Award Documents Language Center Future Shockers	Student Life Accountancy McKnight Center Council of Women Commute Small Business	Dance Gerontology Marketing College Bylaws Why Wichita? Tickets
Geology Manufacturing Management UCATS Alumni News Saso	Intercollegiate Bowling Wichita Gateway Transfer Day Job Openings Live Radio	Thinker & Movers Alumni Foundations Corbin Center Jardine Hall Hugo Wall School	Career Services Doers & Shockers Core Values Grace Wilkie Hall Strategic Plan Medical Tech
Educational Map Physical Plant Graphic Design Non Credit Class Media Relations Advertising	Beta Alpha Psi Liberal Arts Counseling Biological Science Duerksen Fine Art EMT Program	Staff Aerospace Choral Dept. Alberg Hall French Spanish	Softball, Men's McKinley Hall Email Dental Hygiene Tenure Personnel Policies
English Graduate Complex Music Education Advising Center Medical School Levitt Arena	Religion Art Composition Physics Entrepreneurship Koch Arena Roster	Parents Wrestling Philosophy Wichita Lyceum Fairmount Center Women's Museum	Instrumental Nursing Opera Sports History Athletic Dept. Health Plan

Cognitive Aspects of Design

Findings Based on Activity 3 and 4

- Weller (2004) study on grouping and search efficiency found that:
  - People took less time to locate items when information was grouped using a border (Activity 4) compared to using color contrast (Activity 3).
  - The use of visual grouping techniques can enhance the search process and improve efficiency.
- White space and information retrieval:
  - Some argue that excessive white space on web pages can hinder the search process and make it difficult to find information.
  - The impact of white space on search efficiency is a topic of debate.
- Question for discussion:
  - Do you agree that too much white space on web pages is detrimental to the search process?

Perception - Activity 5: Which is the easiest to read and why?

What is the time?

What is the time?

What is the time?

What is the time?

What is the time?

What is the time?

Cognitive Aspects of Design

Perception - Design implications for user interface

- Icons should be designed to ensure users can readily distinguish their meaning.
- Utilize effective visual techniques such as bordering and spacing to group information and improve visual organization.
- Sounds used in the interface should be audible and distinguishable to provide meaningful feedback.
- When designing with colors, research proper color contrast techniques to ensure accessibility and usability.
  - For example, yellow on black or blue is suitable, while yellow on green or white should be avoided.
- Haptic feedback should be used judiciously, providing tactile feedback only when necessary to enhance the user experience.

Memory

- Memory involves the processes of encoding and retrieving knowledge.
- Our memory selectively filters and processes information based on what we attend to.
- Recognition is often better than recall, meaning we can more easily recognize things rather than recalling them from memory.
- Interestingly, research suggests that we remember less about objects we have photographed compared to when we observe them directly with the naked eye (Henkel, 2014).

Memory: Context is Important

- The context in which information is encoded plays a significant role in its subsequent retrieval.
- People may find it challenging to recall information that was encoded in a different context. For example:
  - "You are on a train and someone comes up to you and says hello. You don't recognize him for a few moments but then realize it is one of your neighbours. You are only used to seeing your neighbour in the hallway of your apartment block and seeing him out of context makes him difficult to recognize initially"

Memory: Recognition versus Recall

- Command-based interfaces often require users to recall information from memory, such as recalling a specific name from a large set of options.
- Graphical interfaces, on the other hand, provide visually-based options like menus and icons that users can browse through until they recognize the desired option.
- Web browsers enhance recognition memory by offering features like tabs and history lists of visited URLs, allowing users to easily recognize and revisit previously accessed web pages.

Cognitive Aspects of Design

Memory: The Problem with the Classic ‘7 ± 2’ Theory

George Miller’s (1956) theory of how much information people can remember:

- People’s immediate memory capacity is very limited to 7 plus or minus 2 (7 ± 2).
- This theory has been widely used in interaction design to determine the number of options to display.
- However, its applicability and usefulness in the field of Human-Computer Interaction (HCI) are subject to debate.
- Questions arise regarding whether relying solely on this theory is a good approach in designing user interfaces.

What Some Designers Do: The 'Rule of 7'

- Some designers adhere strictly to the "rule of 7" in UI design:
  - They limit the number of options on a menu, icons on a toolbar, bullets in a list, items on a pull-down menu, or tabs on a website page to only 7.
- However, this approach may not always be appropriate or effective.
- Limiting options to 7 based on this rule can be misleading and oversimplified.
- The "rule of 7" is not universally applicable and may not consider the complexity of the task or the cognitive abilities of the users.

The Limitations of the 'Rule of 7' in UI Design (1/2)

- The 'Rule of 7', which suggests limiting the number of items in an interface to 7 ± 2, may not be ideal in UI design.
- People can easily scan lists, tabs, and menu items to locate the desired option, rather than relying on memory recall.
- This means that interfaces can accommodate more than nine items without overwhelming users.
  - For example, history lists in web browsers can include a larger number of visited websites.
- In some cases, a small number of items can be beneficial, such as on smartwatch displays where screen estate is limited.

The Limitations of the 'Rule of 7' in UI Design (2/2)

- The suitability of the 'Rule of 7' depends on the specific task requirements and the available screen space.
- It is important for designers to consider the context, user preferences, and usability testing when determining the optimal number of items in an interface.
- By blindly adhering to this rule, designers may inadvertently sacrifice functionality, user experience, and efficient navigation.

Digital Content Management: Organizing and Finding Files (1/2)

- With the increasing volume of digital content, users face challenges in managing and locating their files, including documents, images, music, videos, emails, and more.
- Remembering file names and storage locations can be difficult, requiring both recall-directed and recognition-based scanning processes.
- File management systems should be designed to support both memory processes effectively.
- Features like a search box and history list can aid in quickly locating files based on recall or recognition.

Digital Content Management: Organizing and Finding Files (2/2)

- To enhance file organization, users should be provided with options to encode files in richer ways, such as using colors, flags, images, flexible text, and time stamping.
- These features enable users to visually and contextually categorize and remember their files.

Memory Aids for Enhanced Recall (1/2)

- SenseCam, developed by Microsoft Research Labs (now Autographer), is a wearable device that automatically captures photos at intervals without user intervention.
- The digital images captured by SenseCam are stored and can be revisited using specialized software.
- Studies have shown that SenseCam can improve memory, particularly in individuals with dementia.
- Another memory aid is RemArc, which utilizes archived BBC materials to trigger long-term memory.

Cognitive Aspects of Design

Memory Aids for Enhanced Recall (2/2)

- These memory aids offer innovative ways to support and enhance recall abilities.
- By capturing and reviewing visual stimuli, SenseCam helps individuals remember events and experiences that might otherwise be forgotten.

Figure 1: SenseCam Device and Picture Captured

Design Implications for Memory Enhancement (1/2)

- Minimize Cognitive Load: Simplify task procedures to reduce the mental effort required for users to carry out tasks effectively.
- Emphasize Recognition: Design interfaces that prioritize recognition over recall, allowing users to easily identify and locate information without relying heavily on memory.
- Utilize Multiple Labeling Methods: Provide users with diverse options for labeling digital information to enhance recognition and retrieval.
- Use Folders and Categories: Organize files into folders and categories to facilitate efficient organization and navigation.

Cognitive Aspects of Design

Design Implications for Memory Enhancement (2/2)

- Incorporate Color Coding: Apply color to visually differentiate and group related information, making it easier for users to identify and remember specific items.
- Implement Flagging and Bookmarking: To allow users flag or bookmark important content for quick reference and future retrieval.
- Utilize Time Stamping: Automatically record and display time stamps to help users track and recall the chronology of their digital information.
- Consider User Preferences: Provide customization options for labeling methods, allowing users to personalize their organization and retrieval strategies.

Cognitive Aspects of Design

Learning

- Learning involves acquiring skills and knowledge through memory processes.
- Types of Learning:
  - Incidental Learning: Unplanned and spontaneous learning that occurs through everyday experiences and observations.
  - Intentional Learning: Deliberate and purposeful learning, often involving studying or training.
- Various technologies have been developed to support and enhance the learning process.
  - Multimedia, Animations, Virtual Reality, etc.
- People tend to prefer learning through hands-on experiences rather than relying solely on instructional manuals or passive instruction.

Cognitive Aspects of Design

Design implications for Enhancing Learning

- Keep speech-based menus and instructions concise.
- Pay attention to the intonation of artificially generated speech voices, as they may be more challenging to understand than human voices.
- Allow users to increase the text size on the screen to improve readability.
- Design interfaces that encourage exploration and active engagement.
- Provide structured and guided learning experiences through interface design.
- Utilize dynamic linking of concepts and representations to facilitate understanding of complex material.

Reading, speaking, and listening

The ease with which people can read, listen, or speak differs

- Many users prefer auditory information over written text.
- Reading can be faster compared to speaking or listening.
- Listening requires less cognitive effort than reading or speaking.
- Dyslexic individuals may face difficulties in understanding and recognizing written words.

Reading, speaking, and listening - Applications

- Voice user interfaces allow users to interact with them by asking questions
  - E.g. Google Voice, Siri, Alexa
- Speech-output systems use artificially-generated speech
  - E.g. written text-to-speech systems for the visually impaired
- Natural-language systems enable users to type in questions and give text-based responses
  - E.g. chatbots

Cognitive Aspects of Design

Design implications - Reading, speaking, and listening

- Provide options for audio-based content to cater to users' preference for listening.
- Provide options to increase the size of text on the screen, ensuring that users with visual impairments can comfortably read the content.
- Design interfaces that facilitate efficient reading, such as clear typography and appropriate formatting.
- Use concise and accessible language for spoken instructions or text-to-speech functionalities.
- Incorporate features like dyslexia-friendly fonts or customizable reading settings to support users with reading difficulties.

Cognitive Prosthetic Devices: Implications for Designing Learning Technologies

- Increasing reliance on the internet and smartphones for information retrieval reduces the need for extensive memorization.
- The expectation of having internet access impacts our memory by prioritizing knowledge of where to find information online rather than storing it internally (Sparrow et al., 2011).
- Designing technologies to support learning should consider the shift in reliance on external information sources and the implications it has for the learning process.

Cognitive Aspects of Design

Dilemma: Impact of the App Mentality on Decision-Making

- The younger generation's growing reliance on apps is leading to increased risk aversion and difficulty in making independent decisions (Gardner and Davis, 2013).
- The abundance of apps provides ready-made solutions for various tasks, reducing the need for critical thinking and problem-solving skills.
- Relying heavily on apps can lead to anxiety and a lack of confidence in making decisions without app-based guidance.
  - Consider a scenario where a young adult relies heavily on a food delivery app for all their meals. They may become anxious and unsure when faced with the task of cooking a meal from scratch or making dietary choices without the app's suggestions. Any other examples?

Cognitive Frameworks: Understanding User Behavior at the Interface (1/2)

- Cognitive frameworks are utilized to explain and predict user behavior when interacting with interfaces.
- They are based on theories of behavior and focus on the mental processes that occur during interaction.
- These frameworks also consider the role of artifacts and representations in shaping user cognition.

Cognitive Frameworks: Understanding User Behavior at the Interface (2/2)

- Some of the most well-known cognitive frameworks include:
  - Mental models, which depict users' internal representations of how a system works.
  - Gulfs of execution and evaluation, which describe the gaps between users' intentions and the system's feedback. reading assignment
  - Distributed cognition, which explores how cognitive processes are distributed across individuals, artifacts, and the environment. reading assignment
  - External and embodied cognition, which recognize the role of external tools and physical interactions in cognitive processes. reading assignment

Mental Models: Understanding User 'Understanding'

- Users develop an understanding of a system through learning and interacting with it.
- This understanding is often referred to as a mental model, which consists of:
  - Knowledge of how to use the system and what actions to take next.
  - Understanding of how the system works, particularly in unfamiliar or unexpected situations.
- Mental models enable users to make inferences and carry out tasks effectively.

Everyday reasoning and mental models: Activity 6

1. You arrive home on a cold winter's night to a cold house. How do you get the house to warm up as quickly as possible? Set the thermostat to be at its highest or to the desired temperature?
2. You arrive home starving hungry. You look in the fridge and find all that is left is an uncooked pizza. You have an electric oven. Do you warm it up to 100 degrees (Celsius) first and then put it in (as specified by the instructions) or turn the oven up higher to try to warm it up quicker?

Activity 6: Thermostat Control Misconceptions (1/2)

- When asked how to heat up a room or oven that is thermostat-controlled, many people choose the option of increasing the temperature setting.
- This choice is based on the misconception that raising the temperature will heat the room or oven more quickly.
- This misconception stems from the general valve theory, where the "more is more" principle is commonly applied to various settings, such as gas pedals, gas cookers, taps, and radio volume.
- However, this mental model is incorrect when applied to thermostats, which operate based on an on-off switch mode.

Activity 6: Thermostat Control Misconceptions (2/2)

- These misconceptions can be attributed to the general valve theory being generalized across different settings, leading to erroneous mental models.
- Understanding the correct operation of thermostats and educating users about their functioning can help dispel these misconceptions and promote more efficient and effective use of thermostat-controlled systems.

Misconceptions in Understanding Interactive Devices (1/2)

- Users often hold erroneous mental models when it comes to understanding how interactive devices and computers work.
- These mental models are often poor, incomplete, easily confusable, and based on inappropriate analogies and superstition (Norman, 1983).
- For example, when using elevators and pedestrian crossings, many people tend to press the button at least twice.
- This behavior is driven by the misconception that pressing the button multiple times will make the lights change faster or ensure that the elevator arrives promptly.

Misconceptions in Understanding Interactive Devices (2/2)

- Similarly, users may have various other misconceptions about how interactive devices operate.
- These mental models can be influenced by limited knowledge, past experiences, and cultural factors.
- Understanding the kinds of mental models users have is crucial for designing user-friendly interfaces and providing appropriate feedback.

Cognitive Aspects of Design

Designing UX for Better Mental Models (1/2)

- Clear and easy-to-use instructions should be provided to users to help them build accurate mental models.
- Appropriate tutorials and contextual-sensitive guidance can guide users in understanding how to interact with the interface effectively.
- Providing online videos and chatbot windows can offer additional support and assistance when users need help.
- Transparency in interface design is crucial to make the system intuitive to use.
- Interfaces should provide clear affordances that indicate the available actions, such as swiping, clicking, or selecting.

Designing UX for Better Mental Models (2/2)

- Consistency in design elements, terminology, and interactions across the system can help users form consistent mental models.
- User feedback and error messages should be informative and guide users towards the correct mental model.
- Conducting user testing and gathering feedback can identify areas where users struggle to build accurate mental models.
- Continuous improvement and updates to the UX based on user feedback can lead to the development of better mental models.
- Collaborating with users through co-design sessions and involving them in the design process can result in interfaces that align with their mental models.

Human Computer Interaction
CE 382

Course Instructor: Vincent M. Nofong, Ph.D.

July 24, 2024

Introduction

Outline

- Who I am
- Course Information and Outline of CE 382
- Expected Learning Outcomes
- Rules
- Chapter Two: Establishing Requirements

Introduction

About me

- Name: **Vincent M. Nofong, PhD**
- Email: **vnofong@umat.edu.gh**
- Personal Website: https://vincentnofong.com/
- Uni website: https://www.umat.edu.gh/staffinfo/staffDetailed.php?contactID=385
- Office hours (Working days): **09:00 am - 16:00 pm GMT**
- Research interest: **data mining, trend prediction, classification, bioinformatics, artificial intelligence, machine learning**

Introduction

Course Information (CE 382)

- Credit hours: **3**
- Attendance: **10%**
- Continuous Assessment: **30%**
  - Quizzes - two or three
  - Group assignment - one (application development)
  - Group presentations
- End of Semester: **60%**

Introduction

Course Outline (CE 382)

1. Interaction Design
2. Establishing Requirements
3. Prototyping
4. Data Gathering and Analysis
5. Cognitive Aspects of Design
6. Social and Emotional Interactions
7. User Interfaces
8. Evaluations

Introduction

Expected Learning Outcomes (CE 382)

Students should understand and be able to:

1. Explain the characteristics of good and bad interaction design and use them to evaluate HCIs
2. Explain the characteristics of users that influence HCI and use them to inform user interface development
3. Explain, analyze and develop interaction evaluations
4. Explain and develop requirements for interaction design
5. Construct interactions using evaluation-based iterative process for directing the design of user interfaces.

Introduction

Reference Materials

1. Preece, J., Rogers, Y. and Sharp, H. (2023), Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons Ltd, Hoboken, U.S.A., 6th Edition, 716 pp. - slides are based on this reference
2. Lazar, J., Feng, J. H. and Hochheiser, H. (2017), Research Methods in Human-Computer Interaction, Morgan Kaufmann, Burlington, U.S.A., 2nd Edition, 560 pp.
3. Shneiderman B., Plaisant C., Cohen M. and Jacobs, S. (2016), Designing the User Interface, Pearson Publishers, 6th Edition, 616 pp.

Introduction

Rules

1. Feel free to ask questions in class, unless they are too "personal".
2. Students should not be late for lectures or practicals.
3. Students should attend all lectures and practicals.
4. **In case you are unable to attend lectures or will be late, send me an email - at least 30 minutes before lectures.**
5. Students should do and submit all assignments before the given deadline.
6. Unless otherwise permitted, students should not use their mobile phones in class - note usage of Laptops/Desktops is permitted.

HCI CE 382
Chapter Six: Social and Emotional Interactions

Course Instructor: Vincent M. Nofong, Ph.D.

July 24, 2024

Social and Emotional Interactions

Social Interaction in the Digital Age

- Humans are social beings, engaging in various forms of interaction in everyday life.
- Social technologies have emerged to facilitate social connections even when physically apart.
- **Importance of Social Interaction**
  - **Building Relationships**: Social interactions foster connections, trust, and a sense of belonging.
  - **Communication and Collaboration**: Effective communication enhances teamwork and cooperation.
  - **Emotional Well-being**: Social interactions contribute to happiness, support, and emotional fulfillment.

Social and Emotional Interactions

Social Interaction in the Digital Age

- **Negative Impact of Social Technologies:**
  - **Shallow Interactions**: Superficial connections and a decrease in meaningful face-to-face interactions.
  - **Distraction and Addiction**: Excessive use of social technologies can impact productivity and well-being.
  - **Privacy and Security Concerns**: Sharing personal information and potential risks of online interactions.
- Social technologies have transformed the way we interact, offering both opportunities and challenges.
- Finding a balance between online and offline interactions is essential for social well-being.

Social Interaction in the Digital Age

Questions Raised by Social Technologies:

- Are in person conversations being superseded by social media interactions?

Social Interaction in the Digital Age

Questions Raised by Social Technologies:

- Are in person conversations being superseded by social media interactions?
- How many friends do you have on Facebook, LinkedIn, WhatsApp and so on versus real life?
- How much do they overlap?
- How are the ways that we live and interact with one another changing?
- Are the established rules and etiquette still applicable to online and offline?

Social Interaction in the Digital Age

Conversational Mechanisms and Rules (1/5)

- Conversations involve various mechanisms and “rules” that guide the flow of interaction.
- Example of Face-to-Face Conversation:
  - **Mechanism 1: Mutual Greetings**:
    - A: “Hi there”
    - B: “Hi!”
    - C: “Hi”
  - **Mechanism 2: Small Talk**:
    - A: “All right?”
    - C: “Good, how’s it going?”
    - A: “Fine, how are you?”
    - C: “OK”

Social Interaction in the Digital Age

Conversational Mechanisms and Rules (2/5)

- Conversations involve various mechanisms and “rules” that guide the flow of interaction.
- Example of Face-to-Face Conversation:
  - **Mechanism 3: Exchanging Personal Updates:**
    - B: “So-so. How’s life treating you?”

Social Interaction in the Digital Age

Conversational Mechanisms and Rules (3/5)

- Conversational rules provide a framework for understanding how conversations unfold.
- **Rule 1 (Turn-Taking)**: The current speaker chooses the next speaker by asking an opinion, question, or request.
- **Rule 2 (Self-Selection)**: Another person decides to start speaking without being directly prompted.
- **Rule 3 (Continuity)**: The current speaker continues talking until they choose to end their turn.

Social Interaction in the Digital Age

Conversational Mechanisms and Rules (4/5)

- Other conversational rules that help coordinate conversations:
  - **Backchanneling**: Backchanneling involves using verbal and non-verbal cues to signal attention and encourage the speaker to continue. E.g. of backchanneling cues: "Uh-uh", "Umm", "Ahh".
  - **Following**: Following refers to the ability to understand and respond appropriately to the ongoing conversation.
  - **Farewell Rituals**: Farewell rituals are used to signal the end of a conversation and indicate the intention to leave. E.g: "Bye then", "See you", "Yeah, bye", "See you later".

Social and Emotional Interactions

Social Interaction in the Digital Age

Conversational Mechanisms and Rules (5/5)

- Other conversational rules that help coordinate conversations:
  - **Implicit Cues**: Implicit cues are non-verbal signals that indicate a person's readiness to end the conversation. E.g.: looking at the watch, fidgeting with coat and bags.
  - **Explicit Cues:** Explicit cues involve direct verbal communication indicating the desire to end the conversation. E.g.: "Oh dear, look at the time", "I must go", "I'm running late".
- Understanding these conversational rules can help in analyzing and improving communication dynamics.
- Awareness of these rules can enhance conversational flow and avoid interruptions.

Social Interaction in the Digital Age

Breakdowns in Conversation

- Breakdowns in conversation can occur when there is a misunderstanding or miscommunication between speakers.
- When a speaker's statement is misunderstood, they may repeat their statement with emphasis to clarify their intention.
- Tokens are linguistic devices used to indicate confusion or the need for clarification. E.g.: "Eh?", "Huh?", "What?"

Online Conversations: Understanding Dynamics and Breakdowns

- Online conversations have become increasingly prevalent in our digital age.
- Do the same conversational rules apply in online conversations?

Online Conversations: Understanding Dynamics and Breakdowns

- Online conversations have become increasingly prevalent in our digital age.
- Do the same conversational rules apply in online conversations?
- While some rules may still be relevant, the online context introduces new dynamics and norms.
  - **Breakdowns in Email Conversations**: often repaired by providing clarification or using emoticons to convey emotions.
  - **Breakdowns in Instant Messaging**: typically repaired through quick clarifications or using emojis to express emotions.
  - **Breakdowns in Texting**: repaired by sending follow-up messages to correct errors or provide additional context.
  - **Breakdowns in Videoconferencing**: repaired by reconnecting

Social Interaction in the Digital Age

Remote Conversations

- Remote conversations refer to interactions between individuals who are physically separated.
- Various applications have been developed to facilitate remote conversations.
  - E.g.: videoconferencing, instant messaging, and chatrooms.
- Remote conversations offer unique advantages, such as overcoming geographical barriers and enabling real-time communication.
- Despite their benefits, remote conversations also pose challenges, including technological constraints and potential for misinterpretation.

Social Interaction in the Digital Age

Remote Conversations: Early Videophone and Visualphone

Early British Telecom's Videophone

An Early Mobile Visualphone developed in Japan

Remote Conversations: Current Videoconferencing

- Videoconferencing platforms like Teams and Zoom have transformed the way we communicate, offering seamless transitions between conversations and collaborative work.
- However, they also present unique challenges that impact users' experiences.

Social and Emotional Interactions

Remote Conversations: Current Videoconferencing

- Videoconferencing platforms like Teams and Zoom have transformed the way we communicate, offering seamless transitions between conversations and collaborative work.
- However, they also present unique challenges that impact users' experiences.
  - **Zoom Fatigue** (identified by Bailenson (2021)): Excessive amounts of close-up eye gaze during videoconferencing can lead to mental exhaustion and strain.
  - **Cognitive Load**: Videoconferencing requires intense cognitive load, as individuals need to process both verbal and non-verbal cues simultaneously. This increased cognitive demand can contribute to mental fatigue and reduced attention span.

Social Interaction in the Digital Age

Reading Assignment:

- The impact of Facebook and Twitter
- The Skype success
- Coordination Mechanisms
- Awareness Mechanisms
- Designing technologies to support awareness

Not examinable

Social Interaction in the Digital Age

Notification Systems

- Allow users to initiate communication and share relevant information with others, rather than being constantly monitored.
- They provide updates on shared objects and the progress of collaborative tasks, enhancing communication and collaboration within teams.

Social and Emotional Interactions

Social Interaction in the Digital Age

Notification Systems: Benefits

- **User Empowerment**: Users have control over when and how they notify others, promoting autonomy and agency.
- **Information Sharing**: Notification systems facilitate the sharing of updates, changes, and important insights with team members.
- **Task Progress Tracking**: Users can stay informed about the progress of collaborative tasks and have a clear overview of the project's status.

Social and Emotional Interactions

Social Interaction in the Digital Age

Notification Systems: Design Considerations

Social Interaction in the Digital Age

Notification Systems: Design Considerations

- **Clear and Intuitive Interface:** Notification systems should have a user-friendly design that allows for easy navigation and understanding.
- **Customization Options:** Users should have the ability to customize their notification preferences to suit their communication needs.
- **Privacy and Security:** Ensuring the confidentiality of notifications and protecting user data is essential.

Emotions and Behavior in HCI

- Emotional interaction explores how users feel and react while interacting with technologies.
- Affective computing aims to recognize, interpret, and respond to human emotions through technology.
- Understanding emotional responses helps design elements trigger specific emotional reactions in users.
- Designing for happiness, fostering trust, and motivating learning are essential considerations.
- Emotionally responsive interfaces adapt to users' emotional states and provide appropriate feedback.
- User-centered emotional design prioritizes users' needs, preferences, and emotional well-being.

Emotional Interaction in the Digital Age

Emotional Interaction in User Experience

- Emotional interaction involves understanding and addressing human emotions, such as happiness, sadness, annoyance, anxiety, frustration, and motivation.
- Translating emotional knowledge into various aspects of the user experience is essential for impactful design.
- Achieving emotional interaction in design is challenging due to the dynamic nature of human emotions.
- Emotions can be influenced by various factors, making emotional design complex and context-dependent.

Emotional Interaction in the Digital Age

Activity: Understanding Emotions in Online Shopping (1/3)

- Consider your emotions when buying a big ticket item online (e.g., refrigerator, vacation, computer).
- Reflect on the different emotions experienced during an everyday online shopping activity for products like a new phone, washing machine, or vacation.

Emotional Interaction in the Digital Age

Activity: Understanding Emotions in Online Shopping (2/3)

- The process starts with the realization of needing or wanting the item, followed by the desire and anticipation to purchase it.
- Exploring numerous websites, comparison sites, reviews, and recommendations generates joy or frustration during the decision-making process.
- The thrill of selecting a product may be followed by the shock of its cost, leading to disappointment if affordability is an issue.
- Revisiting options and seeking expert advice may evoke annoyance and mistrust in sales assistants.

Emotional Interaction in the Digital Age

Activity: Understanding Emotions in Online Shopping (3/3)

- The struggle continues, causing tiredness and increased frustration during the search for alternatives.
- Relief ensues once a decision is made, but the online payment process can induce anxiety and concern about data accuracy.
- Doubts and second-guessing arise after completing the purchase, causing uncertainty about the chosen product.

Social and Emotional Interactions

Emotional Interaction in the Digital Age

Designing Interfaces to Match and Influence Emotions

- Can interfaces be designed to match or change our emotions? Should they aim to improve how we feel?
- Understanding the continuous fluctuations in our moods and feelings is crucial in interface design.
- How can interfaces keep track of our emotional states and respond appropriately?
- Certain moods may align better with specific interface designs, impacting the overall user experience.
- Designing for various emotions like happiness, anger, sadness, boredom, or focus requires thoughtful consideration.

Social and Emotional Interactions

Designing Interfaces to Match and Influence Emotions: Design Considerations:

- **Emotional Alignment**:
  - Match the interface design with the user's current emotional state.
- **Visual Cues**:
  - Use color, animations, and graphics to evoke specific emotions.
- **Interaction Styles**:
  - Adapt interactions to suit different emotional contexts.
- **Personalization**: Offer customization options to let users tailor the interface based on their preferences.
- **Emotional Feedback**: Provide emotive responses to user actions, reinforcing positive emotions.
- **Mindfulness and Empathy**: Design interfaces that promote well-being and empathy towards users.

Designing Interfaces to Match and Influence Emotions: Challenges

- **Real-time Emotional Tracking**: Implement technology to gauge user emotions accurately.
- **Ethical Considerations**: Ensure respect for user privacy and consent when dealing with emotional data.
- **Cultural Sensitivity**: Emotions can be interpreted differently across cultures; consider inclusivity.
- **Emotional Regulation**: Balance positive emotional influences without manipulating or exploiting users.
- **User Acceptance**: Address concerns about emotional surveillance and manipulation.

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**:

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**: Vibrant colors, playful animations, and positive affirmations.
- **Angry**:

Social and Emotional Interactions

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**: Vibrant colors, playful animations, and positive affirmations.
- **Angry**: Simplified interfaces to minimize frustration and calming elements.
- **Sad**:

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**: Vibrant colors, playful animations, and positive affirmations.
- **Angry**: Simplified interfaces to minimize frustration and calming elements.
- **Sad**: Gentle animations, comforting colors, and supportive content.
- **Bored**:

Social and Emotional Interactions

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**: Vibrant colors, playful animations, and positive affirmations.
- **Angry**: Simplified interfaces to minimize frustration and calming elements.
- **Sad**: Gentle animations, comforting colors, and supportive content.
- **Bored**: Engaging visuals, interactive elements, and surprise features.
- **Focused**:

Designing Interfaces to Match and Influence Emotions: Designing for Specific Emotions

- **Happy**: Vibrant colors, playful animations, and positive affirmations.
- **Angry**: Simplified interfaces to minimize frustration and calming elements.
- **Sad**: Gentle animations, comforting colors, and supportive content.
- **Bored**: Engaging visuals, interactive elements, and surprise features.
- **Focused**: Minimal distractions, clear hierarchy, and task-oriented layouts.

The Complex Relationship between Emotions and Behavior

- Emotions and behavior are interconnected in intricate ways.
- How does anger impact concentration? Does happiness influence risk-taking behaviors?
- Emotions can influence decision-making, cognitive performance, and daily actions.
- Angry individuals may focus better or become more distracted.
- Happiness could lead to increased risk-taking, like spending more money.
- Understanding the complex interplay between emotions and behavior is crucial for human-computer interaction design.

Emotional Interaction in the Digital Age

Emotional Dynamics: Automatic vs. Conscious Emotions

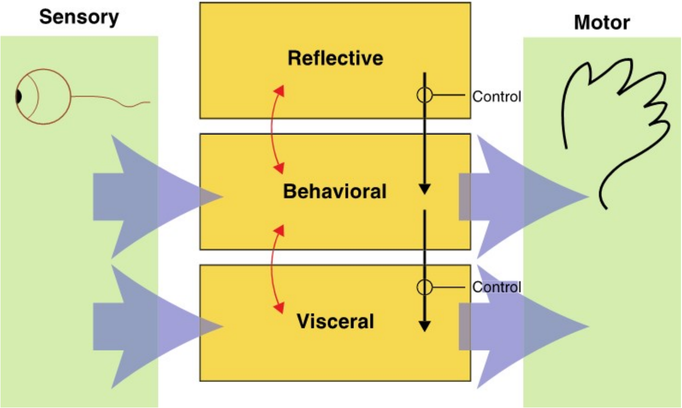
- Emotions vary in duration and complexity.
- They can be categorized into automatic and conscious emotions.
- Automatic emotions are rapid and short-lived, like a sudden fit of anger.
- Conscious emotions develop slowly and endure over time, such as jealousy.
- Automatic emotions dissipate quickly, while conscious emotions linger and may require reflection for resolution.

Social and Emotional Interactions

Emotional Design Model

1. **Visceral Design: Aesthetics and Sensory Appeal**
   - Focuses on making products visually attractive, tactilely pleasing, and appealing to the senses.
   - Emphasizes the first impressions and emotional reactions to the product's appearance, texture, and sound
2. **Behavioral Design: Usability and Functionality**
   - Focuses on the product's ease of use, functionality, and practicality.
   - Aligns with traditional usability principles to ensure efficient and effective interaction.
3. **Reflective Design: Meaning and Personal Value**
   - Explores the emotional and personal significance of the product to the user.
   - Considers the deeper connections, memories, and emotional experiences associated with the product.

Emotional Design Model



Social and Emotional Interactions

Analyzing Swatch Watch Design Using the Emotional Design Model

- Brilliant colors and wild design elements that immediately catch the user's attention -Visceral Design
- Affordances that make it easy for users to interact with the watch and use its features - Behavioral Design
- Cultural images and graphical elements that evoke deeper meanings and personal connections -Reflective Design

Classification of Emotions through Facial Expressions

- Six Core Expressions
  - **Sadness, Disgust, Fear, Anger, Contempt, Joy**
  - Note: These core expressions can also be detected in text as well. How?
  - Reading assignment: Some literature classify emotions beyond the six above Plutchik's wheel of emotions - not examinable
- AI Detection: Facial expressions are measured based on the presence or absence of:
  - Smiling, Eye widening, Brow raising, Brow furrowing, Raising a cheek, Mouth opening, Upper-lip raising, Wrinkling of the nose

Social and Emotional Interactions

Utilizing Emotional Data in Interaction Design

- Adaptive content based on emotional states:
  - Websites can modify ads, movie storylines, or content to match the user's emotional state
- Emotional support and assistance:
  - In a car system, detecting an angry driver may prompt a suggestion to take a deep breath
- Comprehensive data analysis (e.g. sentiment analysis):
  - Emotional data is not limited to facial expressions, but also includes eye-tracking, finger pulse, speech, and textual analysis (e.g., tweets, Facebook posts)
- Reading assignment - How are emotions detected with technology? -Not examinable

Indirect Emotion Detection: Ethical Considerations

- Emotion detection is not only used for direct emotional responses but also to infer behavior. E.g:
  - Assessing a person's suitability for a job or predicting voting behavior in an election or sentiment analysis from tweets.

Ethical Concerns:

- Is it ethical for technology to read emotions from facial expressions or social media posts?
- Considerations for privacy and data usage
- Potential biases and accuracy issues in emotion detection algorithms
- Balancing the benefits of personalized experiences with user consent and data protection.

Anthropomorphism in Human-Computer Interaction

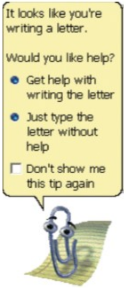
- Attributing human-like qualities to non-human objects e.g. cars, computers, household items, etc.
- Commonly used in advertising: dancing butter, animated drinks, talking breakfast cereals, etc.
- Leveraging anthropomorphism to create engaging and relatable interfaces
- Providing a sense of connection and emotional attachment to technology

The Power of User Feedback in Interface Design

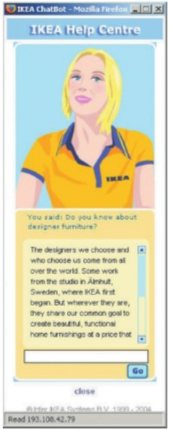
- **Welcome Message**:
  - Option 1: "Hello Chris! Nice to see you again. Welcome back. Now, what were we doing last time? Oh yes, Exercise 5. Let's start again."
  - Option 2: "User 24, commence Exercise 5."
  - Which message do you prefer?
- **Feedback on Incorrect Answers**:
  - Option 3: "Now Chris, that's not right. You can do better than that. Try again"
  - Option 4: "Incorrect. Try again"
  - Which message do you prefer?

Social and Emotional Interactions

The Rise and Fall of Virtual Agents: A Tale of Clippy and Anna



(a)



(b)

The Rise and Fall of Virtual Agents: A Tale of Clippy and Anna

Microsoft's Clippy - A Well-Intentioned Desktop Agent

- Introduced as a helpful desktop assistant by Microsoft
- Intended to enhance user experience and aid productivity
- However, Clippy's reception was far from positive
- Clippy faced widespread dislike among users
- Perceived as annoying, distracting, and patronizing
- Its presence often interfered with users' tasks and workflow

The Rise and Fall of Virtual Agents: A Tale of Clippy and Anna

IKEA's Anna - A Virtual Agent with Facial Expressions

- IKEA introduced Anna as a virtual agent to assist customers
- Utilized blinking, moving lips, and head gestures for facial expressions
- Aimed to provide a more interactive and engaging experience

The Rise and Fall of Virtual Agents

Lessons learned from Clippy and Anna

- User preferences and receptiveness matter
- Balancing helpfulness and intrusiveness is crucial
- Implementing subtle and contextually relevant animations can improve user engagement

Designing the Future of Virtual Agents

- Leveraging AI advancements for more intelligent and empathetic virtual agents
- Prioritizing user feedback and preferences to create delightful interactions
- Striking a balance between assistance and user autonomy

Crafting Effective Error Messages: Shneiderman's Guidelines

"The application Word Wonder has unexpectedly quit due to a type 2 error."

Crafting Effective Error Messages: Shneiderman's Guidelines

"The application Word Wonder has unexpectedly quit due to a type 2 error."

Why not instead:

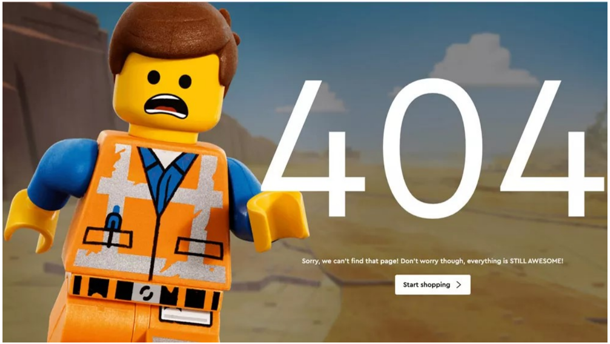
"The application has expectedly quit due to poor coding in the operating system"

Shneiderman's classic guidelines for error messages include:

- Avoid using terms like FATAL, INVALID, or BAD
- Audio warnings
- Avoid UPPERCASE and long code numbers
- Messages should be precise rather than vague
- Provide context-sensitive help

Social and Emotional Interactions

A funny image incorporated into a 404 error message



What are the advantages and disadvantages of the above error message?

The Power of Anthropomorphism in Educational Software

- Reeves and Naas (1996) discovered the impact of anthropomorphism in computer interactions.
- Computers that offer flattery and praise in educational software:
  - Result in positive user experiences.
  - Boost engagement and motivation.
  - Encourage students to continue with exercises.
- Anthropomorphism enhances user engagement in educational software.
- Incorporating personalized and positive feedback:
  - Increases user satisfaction.
  - Encourages active participation.
  - Fosters a positive learning environment.

Criticism of Anthropomorphism in User Interfaces

- **Deceptive**: Anthropomorphism may give users false expectations, leading to disappointment or frustration.
- **Anxiety** & **Inferiority**: Human-like characters can make users feel judged/inferior, affecting their confidence and motivation.
- **Impersonal vs. Personal Feedback**: Users may prefer more impersonal feedback, such as "Incorrect. Try again." over character-driven feedback like "Now Chris, that's not right. You can do better than that. Try again."
- **Balance in Design**: Careful consideration of anthropomorphism is necessary to create positive user experiences while avoiding negative impacts.

Social and Emotional Interactions

Dilemma: Should voice assistants teach kids good manners

- Kids interact with voice assistants like Alexa as if they were their friends, often neglecting to use politeness or manners.
- Children may not learn the importance of saying please and thank you, affecting their interactions with technology and potentially transferring to real-life situations.
- **The responsibility lies with parents** and voice assistants to teach good manners and appropriate behavior.
- Note: Recent Research: Studies show that kids differentiate between how they treat voice assistants and how they interact with humans (Alexis Hiniker et al., 2021).

Social and Emotional Interactions

Dilemma: Should computers say they're sorry?

- Would users be as forgiving of computer apologies as they are of each other's apologies?
- How sincere would users perceive the computer's apology after a system crash? E.g.: "I'm really sorry I crashed. I'll try not to do it again."
- Incorporating human-like manners in computer responses may enhance user experience, but sincerity and authenticity should also be considered to avoid potential user skepticism.

Creating Robotic Pets: Design Considerations

- Should robots be plastic-pet-like, cuddly-pet-like, or plastic-human-like?
- Most people enjoy interacting with pets and cuddly toys, making pet-like robots a popular choice.
- While realistic features enhance the emotional experience, some may find human-like robots creepy.

Note:

- Designing robotic pets involves a trade-off between realism and potential discomfort.
- Consider users' emotional experiences and comfort levels to strike the right balance in creating pet-like robotic companions.

Social and Emotional Interactions

Frustrating Interfaces (1/2)

Many causes:

- Application Issues:
  - Application malfunctions or crashes lead to user frustration.
  - System fails to execute users' intended actions.
- Unmet Expectations:
  - User expectations not fulfilled, leading to dissatisfaction.
  - Lack of information hampers users' understanding of the interface.
- Vague and Obtrusive Error Messages:
  - Unclear, obtuse, or condemning error messages worsen frustration.
  - Users struggle to grasp the problem and find a resolution.

Frustrating Interfaces (2/2)

Many causes:

- Design Elements:
  - Garish, noisy, gimmicky, or patronizing interface aesthetics evoke frustration.
  - Aesthetic choices impact user experience negatively.
- Cumbersome Processes:
  - Lengthy and complex task procedures lead to irritation.
  - Discovery of mistakes requiring a restart amplifies frustration.