ILLINOIS INSTITUTE OF TECHNOLOGY

Transforming Lives. Inventing the Future.

www.iit.edu

SOFTWARE ENGINEERING CS 487

Prof. Dennis Hood

Computer Science

Lecture 10
SA andUCD

Lesson Overview

- Situation Awareness: Human-Computer Interface and User-Centered Design
- Reading:
  - Case Study – Autonomous Vehicles and Traffic Laws (posted)
- Objectives
  - Revisit design of the user interface with particular focus on protocols for supporting the interaction between the human user and the computer application (HCI)
  - Examine the ease-of-use non-functional requirement with particular focus on user-centered design (UCD) and situation awareness (SA)

Participation – P9

- Describe the alarm/alert design of a mission-critical automated system of your choosing:
  - How does the system become aware of the "alarming" situation?
  - How does is make humans aware of the situation?
  - Does it assess the awareness of the humans?, and if so, how?, and if not, is that OK?
- Describe the "costs" associated with UCD and use risk assessment to show that they are "worth it".
- Discuss the connection between user categorization and UCD.

Case Study: Autonomous Vehicles and Traffic Laws

- Requirement – the vehicle must follow the law
  - This actually represents a set of requirements such as “the vehicle must not exceed the speed limit”
- Potential problems
  - The law may be ambiguous
  - Laws may conflict
  - Human-driven vehicles may not obey the law
  - Exceptional conditions may require breaking the law to avoid a more serious problem

Give the People What They Want

- The user’s opinion is more important than yours
- Appreciate users’ capabilities
- Offer help – in many forms
- Strive for quality user experiences
- Involve the user early
- Use what works
- Things can be interpreted differently by different people at different times

The Design Process

- Identify user groups
- Establish requirements for the user experience
- Develop alternative designs
- Create interactive prototypes
- Evaluate with relevant, objective analysis
- Collect feedback and fix it in the next version

Usability Goals

- Effective to use
- Efficient to use
- Safe to use
- Having good utility
- Easy to learn
- Easy to remember how to use

Design Principles

- Make it visible, or not, as needed
- Provide feedback
- Set boundaries (constraints)
- Be consistent
- Provide clues

Evaluation Described

- Evaluation is the process of assessing the goodness/acceptability of a design
- Why
  - Evaluation is critical to achieving user acceptance
  - It is beneficial to get it right the first time
- What
  - Criteria should match the users' needs/interests
  - Easy to learn, fast, satisfying, entertaining, etc.
- Where
  - My place or yours? Control vs. comfort
- When
  - You can evaluate at any point, but should you?
  - Does it meet standards? Does it meet user's needs?

Evaluation Approaches

- Usability testing
  - Quantification of user performance
    - Time to complete, error rates, type/severity of errors, etc.
  - Measure *typical* users' performance on *typical* tasks
  - Controlled by the evaluator
- Field studies
  - See how users act and interact (with each other, a given product, etc.) in their *natural* setting
- Analytical evaluation
  - Heuristic (guidelines and standards) evaluations
  - Walkthroughs of scenarios using prototypes
- Hybrids

Evaluation Methods

- Observation and inquisition
  - Observe users
  - Ask users, ask experts
  - Observation, questionnaires, interviews
- User testing
  - E.g., based on scenarios
- Inspections
  - E.g., based on heuristics
- Modeling
  - To predict and establish benchmarks

Usability Testing

- Testing the product
  - to determine the extent to which it is usable
  - by the intended user population
  - on the tasks for which it was designed
- User testing
  - Measures human performance on specific tasks
  - Logging of keystrokes and mouse movements, video recordings, etc.
- User satisfaction questionnaires / interviews
  - How do you feel?
  - Efficiency and effectiveness

Measuring Usability

- Time
  - Time to complete a defined task
  - After a specified time away from the product
- Number
  - Number and type of errors made per task
  - Number of errors per unit time
  - Number of navigations to help
  - Number of users making a particular error
  - Number of users completing successfully

User-Centered Design

- Decision support
  - Traditionally technology-centered
  - Sensors, reports, gauges, alarms, etc.
- vs. Information overload
  - Bottlenecks
  - 7 +/- 2 chunks
  - Recall time, processing time
- Operator error
  - A causal factor of 60% to 85% of all accidents
- UCD is improving performance and acceptance

UCD and Situation Awareness

- UCD is not
  - Giving users everything they ask for
  - Making decisions for them
  - Doing things for them
- Design principles
  - Organize technology around the user's needs and capabilities
  - And around the way they process information and make decisions
  - Keep the user in control and aware of state
    - thereby reducing anxiety and
    - improving decision-making effectiveness

Responsibilities of the Designer

- Users of your product should enjoy the experience
- The product should undeniably ease / improve the user’s life
  - Increased productivity
  - Reduced anxiety / minimal frustration
  - Better performance / results
  - Consistency, integrity, reliability, etc.
- The product should exceed all competing products in these areas
  - Customers will be willing to pay more
  - Customers will buy variant products
  - Customers will tell their friends good things

SA's Importance

- SA improves decision making
  - Faster
  - Better
- SA facilitates appropriate automation
  - "normal" is predictable and can therefore be automated safely
- SA better prepares the user for "exceptional"
  - The universe of exceptions is smaller and narrower and is therefore more manageable

SA Levels

- Level 1 – Perception
  - Status, attributes, and dynamics of relevant elements
- Level 2 – Comprehension
  - What does the data mean in the context of relevant goals and objectives
- Level 3 – Projection
  - Predicting future state

Assessment Elements

- Time
  - How long does it take to become situational-ly aware
- A product of the process
  - Users become situational-ly aware through a process of investigation and analysis
- Perception and attention
  - Users have limited capacities for both conscious and sub-conscious processing
  - Context, previous experiences, time, etc.
- Working memory
  - Have I seen / experienced this before?
  - 7 +/- 2 chunks of information in “working” memory
- Mental models
  - Users organize experiences and analyze situations methodically using models proven to be effective
- User goals
  - “Did I get my job done?”

Dashboards

- Information of all kinds can be consolidated into dashboard imagery
- Dashboards have become quite popular
  - Does that mean they are effective?
  - Will popularity help make them effective?
- Historical precedence
  - Car dashboards
  - Industrial control panels
- Business intelligence – so simple a CEO can use it
- Enron fallout – liability of not conveying knowledge to shareholders

Sources of Uncertainty

- SA uncertainty
  - Perception of elements (level 1)
    - Missing (e.g., hidden) information
    - Reliability/credibility of data
    - Conflicting data
    - Timeliness
    - Noisy data
  - Comprehension confidence (level 2)
    - Challenging abstraction
  - Will decisions lead to desired outcomes? (level 3)
- Decision uncertainty is a factor of the uncertainty the user has across the 3 levels of SA

Managing Uncertainty

- Users naturally work to remove uncertainty
  - Rely on defaults
  - Perfection may not be necessary
  - Process of elimination
- Good design helps by
  - Providing supplemental information
  - Representing critical information in multiple ways
  - Provide multiple sources to help resolve conflicts
  - Supporting “bet-hedging”

Support Uncertainty Management

- Make it easy for users to assess reliability
  - Deciding among several close alternatives takes time
  - Fewer, more distinct choices means more rapid decision making
- Cues to assist should be prominently presented proximally to the display information
  - Confirming or negating information
  - Contextual information

Complexity and SA

- Complexity inherently impedes SA
- However complexity is dynamic
  - The more we learn about a complex environment, the simpler it becomes
  - Of course, the demand for more features increases the complexity again
- Designers' goals
  - Balance features and complexity
  - Help users learn and get comfortable
  - Expand the feature set judiciously

System Complexity

- System complexity is relatively objective
  - System complexity can be measured
  - And therefore reduced in a measurable fashion
- Metrics
  - Number of items
    - Objects, functions, etc.
  - Degree of interaction among items
  - System dynamics
    - Frequency of status change
  - Predictability of changes

Operational Complexity

- How difficult is the system to use?
- System complexity does not imply operational complexity (e.g., cars)
  - Reducing operational complexity can offset and even override system complexity
- Each user category will have its own rating of operation complexity
- Automation can reduce operational complexity, but not necessarily

Apparent Complexity

- Apparent complexity is relatively subjective
  - Does the system “look” complex?
  - Basically an assessment of the UI
  - A bad UI can make a simple system complex
- A function of
  - Cognitive complexity
    - Ease of accurately mapping interface to operation
    - Transparency or surfacing system status
  - Display complexity (density, grouping, etc.)
  - Task or response complexity
    - Challenge in accomplishing goals

Complexity Design Principles

- Resist feature creep
- Manage feature group (prioritize)
- Be consistent
- Simplify flow (minimize branching)
- Analogies and metaphors
- Transparency and observability
- Group information logically
- Balance display density with coherence
- Establish and respect standards
- Minimize task complexity

The Role of Alarms

- Sometimes it's really, really important to get the user's attention
  - Something has gone wrong
  - An automated system is no longer "comfortable" being in charge
  - It's time to change course, check the latest readings, etc.
- When it's one of those times, the alarm needs to effectively grab the user's attention
  - Flashing lights
  - Annoying sounds
  - Changing colors, shapes, etc.
  - Multiple approaches may be required

Failure to Alarm

- Too many alarms
  - Information overload / confusion
  - Hard to process => easy to ignore
- False alarms
  - Users react with complacency or even ignore the alarms (crying wolf)
- Intentional disarming
  - Users sometimes disable alarms
  - Often due to #1 and #2 above
  - Responding to “unnecessary” alarms can cost valuable time
- We need better designed alarm systems!
  - Thresholds, triggers, sensitivity, etc.

Alarm Realities

- Studies show that in practice user’s often fail to respond immediately
- Reliability
  - User’s perception of past correlation between the alarm and its value
  - Designers must avoid creating “wolf-crying” systems
    - Be sensitive to environmental specifics
- Users look for confirmation
  - Supporting data, a second alarm, visual cues, etc.
- Expectations and perceptions
  - User’s can become trained to expect alarms under certain circumstances
  - However, the “expected” alarm may have a different cause (than what the user perceives)

Alarm Realities (cont.)

- Disruptions and diversions
  - Clearing an alarm may take the user away from "more important" tasks
- Workload
  - Underload vs. overload, real vs. false alarms
- Alarm characteristics
  - Volume, brightness, frequency, color, melodic structure, etc.
- Diagnosis of alarms
  - Causal analysis vs. preconceived notions
- Alarm reduction schemes
  - E.g., don't alarm during start-up

Design Principles

- Provide support for projection
- Provide support for confirmation
- Remove ambiguity
- Reduce false alarms
- Set trade-offs appropriately
- Use multiple modalities, consistently
- Minimize disruptions
- Support assessment and diagnosis of multiple alarms
- Support global SA

Keep It Simple

- Again, working memory is limited – designing with the limitations in mind is critical
- Less is more (slick, cool, flashy are not)
  - Avoid clutter
  - Avoid unnecessary marks, elements, etc.
  - Avoid scrolling, paging, etc.
  - Every bit of ink (pixel) requires a reason
- Organization is key
  - Group logically and summarize where possible
  - Focus on context
  - Rely on iconic memory where possible

Building Better Automation

- Adaptive automation
  - Periodically pulls the user in
  - When to engage the user
    - When the user seems dis-engaged
    - During critical events
- Levels of automation
  - A more appropriate mix of system and user control

Design Principles for Automation

- Only if necessary
- Routine actions, not high-level tasks
- SA support, not decisions
- Keep the user in-the-loop
- Minimize modes of automation
- Make state obvious
- Enforce consistency
- Avoid advanced queuing of tasks
- Avoid information cueing
- Human/system symbiosis
- Provide transparency

Benefits of Uninhabited Vehicles

- Boldly go where no one can go
  - Distance
  - Time
  - Oxygen
  - Pressure
- Challenging, even dangerous environments
  - Unstable terrain/buildings
  - Combat areas
- Maintaining secrecy and safety
  - Distance from the enemy
- Automation

Challenges

- Somebody still has to drive the thing
  - Start/stop, left/right, up/down, etc.
  - Commands transmitted via specified medium
- The operator is acting without true sensory feedback and simulated feedback has many shortcomings
  - Sluggish response to commands
  - Delays in receiving feedback
  - Missing or misinformation
  - Lack of “big picture”
  - Lack of subtle cues (or at least delay)
- Relatively immature technology
- Absence of risk can lead to carelessness