CMPE 131/231  DANM 231:
Human-Computer
Interaction
Instructor: Sri Kurniawan – E2/331
TA: Alexandra Holloway

Course contact:
srikur@soe.ucsc.edu
fire@soe.ucsc.edu

0.1 What’s this course about?
► Development of user interfaces that are:
  • fit for the purposes
  • of their diverse users
  • in a variety of contexts
► Interactive system development lifecycle
  • gather user requirements
  • create prototypes
  • conduct evaluations to verify the design
► Movie clips of relevant examples of work in the area
► Demonstrations and hands-on exercises on various techniques
► Guest lectures

0.2 Assessment
► For CMPE 131 takers: 2 homework @ 10% = 20%
► For CMPE/DANM 231: 2 homework + reviewing work = 20% (pick 3 HCI papers that interest you, critically analyze and summarize them).
► Exam: 20%
► Group project (3-4 people): 60%
  • The TA will explain more after the break

0.4 Using course resources
► The lecture notes
  • http://www.soe.ucsc.edu/classes/cmpe131/Winter11/
► Suggested readings:
  • Preece, Sharp, Rogers: Interaction Design (2nd ed)
► For discussion and project deliverables
  • Moodle (http://moodle.soe.ucsc.edu)
► If you have any questions:
  1. Ask me/TA questions at the end of the class
  2. Send me/TA an email
  3. In emergency situation, knock on my/her door

0.5 What is HCI?
► A discipline concerned with
  • the design, evaluation and implementation
  • of interactive computing systems for human use and with
  • the study of major phenomena surrounding them
► Design of interactive systems that are:
  • enjoyable to use, that do useful things and that enhance the lives of the people that use them.
  • accessible, usable and engaging.
► Methods for:
  • capturing what people want to do rather than just what the technology/designer can do
  • understanding how to translate from what people wants to good design
  • involving people in the design process
  • designing for diverse users and uses
0.7 Bad designs are everywhere

DZone Spy - Fresh Links In Real Time

0.8 How to avoid bad design?
► Early focus on users and tasks
  ▪ directly studying cognitive, behavioral, anthropomorphic & attitudinal characteristics
► Empirical measurement
  ▪ users’ reactions and performance are observed, recorded and analysed throughout design process
► Iterative design
  ▪ Develop prototypes of increasing complexity that get verified and approved by users. Be ready to throw away!
► In summary:
  ▪ Ideally, users should be involved throughout the process

1 Different Design Approaches
► System-centered design
  ▪ What can be built easily on this platform?
  ▪ What can I create from the available tools?
  ▪ What do I (programmer) find interesting to do?
► Task-centered design
  ▪ Structured around specific tasks the user will want to accomplish with the system being developed.
  ▪ Able to get users but not throughout the process.
  1. Phase 1: Identification: identify users and tasks
  2. Phase 2: Requirements: prioritizing tasks and users
  3. Phase 3: Design: Scenario-based design
  4. Phase 4: Walkthrough Evaluations
► Goal-centered design
  ▪ Articulates users’ goal rather than tasks
  ▪ Tasks change with the technology and can contradict goals. Goals seldom change
  ▪ May result in different task / task sequence which could be better
► User-centered design
  ▪ Design is based upon a user’s abilities and real needs, context, work, tasks; users are involved throughout
► Participatory design (one of UCD) → Scandinavian, union-agreed model
  ▪ Users are active collaborators in the design process
  ▪ Users are considered subject matter experts

2.1. ISO 13407 – Human-centred design process

2.1. User-Centered Design

Up side of UCD
► Users are excellent at reacting/commenting
► Got to know what users want early on
► Users bring in important “folk” knowledge of use context
► Greater buy-in for the system often results

Down side of UCD
► Hard to get a good pool of users (cost, reluctance, timing)
► Can’t expect users to come up with design ideas from scratch
► Users are not always right
► Users don’t always know what they want
► Conflicts - language, cultural and knowledge differences
2.1 Traditional design sequence

- **requirement**: what is wanted
- *interviews* ethnomethodology
- what is there vs. what is wanted
- *analysis*
  - guidelines
  - principles
  - precise specification
- *design*
  - prototyping
- *implement and deploy*
  - architectures
documentation help

2.2 Star method of UCD

- **Implementation**
- **Task / functional analysis**
- **Prototyping**
- **Evaluation**
- **Conceptual / formal design**
- **Requirements specification**

2.2. Star method

► Components:
  - Requirements - finding out what people need from the system
  - Evaluation - checking that you've got it right
  - Conceptual design - creating the overall idea for the new system
  - Physical design - filling in the details of what the new system will be like and how it will work
  - Prototyping and envisionment - bringing ideas to life

► Features of star method:
  - Evaluation is central to everything
  - Activities can happen in 'non-orderly' manner

2.2. Re-decorating your room

► Requirements - I need a space to work in. I want to get rid of some clutter. I want the room to be lighter, fresher, cleaner...

► Conceptual design - need to create an area for working in; need to build a cupboard to store things in; paint the walls a lighter color.

► Physical design - a partition up in that corner; that cupboard I saw in Ikea could be used to store things in my flat; I am going to paint the walls 'apple-white'

► Envisionment - look at this model room in this magazine; here's a sketch of my ideas for a cupboard; you know the colour of Rod's bedroom...

► Evaluate - that partition would be too expensive, that cupboard would get in the way; light-colored wall would get dirty very quickly...

2.3. Requirements: Why

► To understand what we are going to be doing
► We build systems for others, not for ourselves
► Requirements definition: the stage where failure occurs most commonly
► Getting requirements right is crucial

► Understanding problem space
  - Are there problems with the existing product or way people do things?
  - Why do you think there are problems?
  - How do you think your proposed design ideas might overcome these?
  - How will the proposed design extend or change current products or ways of doing things?

2.3. Requirements: Development costs

- Improper requirements analysis: 29%
- Improper software design: 19%
- Improper software specification: 17%
- Improper test on actual machine: 7%
- Modification of hardware specification: 7%
- Others: 16%
- Coding error: 5%

(Japanese METIA, 2004)
2.3. Requirements

Functional:
► Historically requirements
► Features, functions that the system can/cannot do

Non-functional
► ‘The other issues’
► “-ilities” (quality, accessibility, evolveability, flexibility, etc.)
► Constraints
► Cost, aesthetics, etc.
► Usability requirements

2.3. Typical Real-World NF Requirements

► Elapsed time to market
► Cost/effort to design and implement
► Size/footprint/weight/power/price
► Computer power/memory (related to cost and power)
► Consistency with overall product line
► Backward compatibility
► Differentiation from competitive products

2.3. NF Requirements

► Accessibility = ensuring that people with special needs are not at a disadvantage.
► Usability = quality of the interaction in terms of parameters such as time taken to perform tasks, number of errors made and the time to become a competent user.
► Acceptability = fitness for purpose in the context of use. It also covers personal preferences that contribute to users ‘taking to’ an artefact, or not.
► Engagement = designing for great, exciting and riveting experiences.

2.3. NF Requirement: Usability

► Basic ideas: humans are emotional, are not interested in putting a lot of effort into, and generally prefer things that are easy to do vs. those that are hard to do. (McQuillen, 2003).
► ISO 9241 defines usability to consist of three components:
  ▪ effectiveness: the accuracy and completeness with which specified users can achieve specified goals in particular environments
  ▪ efficiency: the resources expended in relation to the accuracy and completeness of goals achieved
  ▪ satisfaction: the comfort and acceptability of the work system to its users and other people affected by its use.

2.3. Requirements: Steps

1. Gather data
   ▪ Interviews, observation, ethnographic study, surveys/questionnaires, documentation, immersion, contextual inquiry, etc
2. Organize data
   ▪ Notes, cards, affinity diagrams, recording, brainstorming, computer tools
3. Represent data
   ▪ Lists, outlines, matrices
   ▪ Scenarios, personas, storyboards, use cases

2.3.1. Requirement gathering: steps

1. General questions/specifications
2. Selecting relevant site(s) and subjects
3. Collection of relevant data
4. Interpretation of data
5. Conceptual and theoretical work
6. Writing up findings/formal specs

5a. Tighter questions/specifications
5b. Collection of further data
2.3.1 Requirements: Gather
► Competitive analysis ➔ good and bad ideas
► Investigate: Activities and Artifacts (not just artifacts)
► Study related processes and objects in the environment that people may use
  ▪ Office environment - papers, whiteboards, ...
  ▪ Phone calling - phone book, note pad, dial, ...
► Focus on observable behaviors
  ▪ What are the practices, methods, steps, objects, ..., used?
► Learn what users do, why they do it, how they do it, when and where they do it, with what tools or people they do it

P.1 Interviews
► Structured interview
  ▪ Replicable but may lack richness
  ▪ Questions with a set of pre-defined answers
  ▪ Order and wording are important
  ▪ Surveys
► Open Qualitative interviews
  ▪ Qualitative analysis
  ▪ Questions are used to start conversations with informants
  ▪ Unstructured - are not directed by a script. Rich but not replicable
  ▪ Semi-structured - guided by a script but interesting issues can be explored in more depth. Can provide a good balance between richness and replicability

P.2 Focus Group
► A group discussion/interview to access memories, feelings and perceptions concerning a specific focused topic
► Typically 7-10 people – small enough for everyone to have the opportunity to share insights; large enough to provide diversity of perceptions. See this movie.
► Moderated by a trained, neutral interviewer (i.e., moderator, facilitator), who asks questions, listens, keeps the conversation on track, and makes sure everyone has a chance to speak
► Must be held in a comfortable, permissive, non-threatening environment
► Three focus groups are considered to be the minimum for a quality study

P.3 Ethnographic Observation
1. A question or concern is identified for study
2. A group to study is identified (typically small and purposively selected) ➔ permission is asked
3. The researcher observes the group
  ▪ Privileged observer – just observes
  ▪ Participant observer – functions as part of the group
4. Researcher watches and listens attentively and records as much detail as possible (this is called naturalistic observation). Large amounts of notes are typically generated.
5. Repeat as long as necessary (a week or years).
6. The researcher analyzes the notes, identifies themes, looks for answers to questions, and makes logical inferences.

P.4 Contextual Inquiry
► Context: go to the users’ workplace and watch them do their own work.
  ▪ Allows the analyst to experience the rich work rather than the impoverished account of a summary
► Partnership: discuss about their work and engage them in uncovering unarticulated aspects of work.
► Interpretation: develop a shared understanding with user about the aspects of work that matter.
  ▪ Sharing interpretations with users is an important test of our reasoning - and they may fine-tune/correct details
► Focus: direct the inquiry from a clear understanding of your own purpose.
  ▪ Focusing on things that are important/interesting

P.5 Questionnaires
► Can be administered to large populations
► Paper, email and the web used for dissemination
► Questions can be closed or open
► Closed questions are easier to analyze, and may be done by computer
  ▪ Dichotomous: offering two choices (yes/no)
  ▪ Multiple choice: three or more
  ▪ Likert Scale: balanced units (“scaled”)
  ▪ Odd # allows neutral stance
  ▪ Even # forces choice ➔ nobody is really neutral
  ▪ Respondent chooses amount of agreement/disagreement (usually 5-10 choices)
► http://www.socialresearchmethods.net/kb/scallik.php
2.3.3. Req. Represent

- **Personas**
  - a set of stereotypical narratives of potential users of the system
- **Scenarios**
  - an informal narrative story, simple, ‘natural’, personal, not generalizable
- **Use cases**
  - assume interaction with a system
  - assume detailed understanding of the interaction
- **Essential use cases**
  - abstract away from the details
  - technology neutral

2.3.3. Req. Represent: Personas

- A model of key user attributes and goals
- Distilled from observing/interviewing real people
- Presented as a vivid, narrative description
- Of a single “person” who represents a user segment
- Used to guide the design of products, channels, and messaging
- Personas need to have goals (what they are trying to achieve w. the product), inclinations, capabilities.
- Personas represent behavior patterns, not job descriptions
- It is best to develop a few (not too many) concrete personas who have hard characteristics such as name, computer experience, etc
- Try to bring the character alive - perhaps include a picture or two

2.3.3 Reasons for Personas

- There is no such thing as an average user
- A compromise design pleases no-one
  - The broader you aim, the more likely you miss the bulls-eye
  - 50% of the people 50% happy doesn’t work: car: soccer mom, carpenter, dot-com exec
  - “Every time you extend functionality to include another constituency, you put another speed bump of features and controls across every other user’s road.”
- A targeted design can achieve
  - 10% people 100% ecstatic
  - Examples: pickup truck, paro

2.3.3 A family of AOL users

- **AI**, 47, Software developer, visually impaired, wants fast connection
- **Mary**, 45, architect, needs GB data transfers, novice Internet user
- **Charlotte**, 16, Nursing student, chats for 2 hrs per day
- **Jane**, 2, loves Sesame Street online games
- **Rufus**, 10, dog, barks on every animations

2.3.3 Sample persona

- **Mary Jones**, 45, architect for SCArc, an urban architecture company based in Santa Cruz
- She took Computing and Internet class at Cabrillo College last Fall. When working at home, she normally starts working at 09:00, taking a break 12:00-14:00 to pick up the children, then go back to work until 18:00. Have to handover computer to Charlotte at 18:00 sharp.
- Most often her remote work involves discussing with clients over skype, sending sketches in jpg (typical size 1GB per file), and updating her office on the discussions.
- **Focus on enabling design decisions**
  - Computer
  - Impatient when file transfer is slow
  - Bad at multitasking (cannot send file and chat at the same time)
- **Key goals**
  - Sending 1GB in less than 1 minute
  - Reliable connection, especially for video chat
  - Secure connection

2.3.4. Req: Represent: Scenarios

- Scenarios are stories about people undertaking activities using technologies in contexts
- Most often narrative but can be complemented with pictures
- Develop conceptual scenarios that cover the main activities that the technology has to support
  - Pete logs onto the computer
- Develop concrete versions of these for specific designs of the technology
  - Pete clicks on the **key** icon in the **File** toolbar
2.3.4 Example of a scenario

1. The user selects Add a Note from the menu. A new window appears.
2. From the list box at the top of the window she selects the name of the client.
3. A list of campaigns appears in the list box below, and she selects a particular campaign.
4. A list of adverts appears in the next list box, and she selects a specific advert.
5. She types a few paragraphs into a text box to describe her idea for the advert.
6. She fills the space on screen and a vertical scrollbar appears and the text in the text box scrolls up.
7. She enters her initials into a text box, and the system checks that she is allocated to work on that campaign.
8. The date and time are displayed by the system, and the Save button is enabled.
9. She clicks on the Save button and the word Saved appears in the status bar.
10. The text box, the text field for initials and the date and time are cleared.

2.3.5. Use Cases

► Interaction between a user (actor) and a given or assumed user interface
► Encompass a set of usage scenarios, bound to the same goal of the primary actor
► Organized into a Main Success Scenario and a set of Extensions
► Contain attributes such as goal, primary actor, precondition, level of abstraction
► Essential use case: A simplified, abstract, generalized use case defined in terms of user intentions and system responsibilities
  - no technological constraint

2.3.5 Essential Use Case

<table>
<thead>
<tr>
<th>User Intention</th>
<th>System Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student identifies himself</td>
<td>Verifies eligibility to enroll</td>
</tr>
<tr>
<td></td>
<td>Indicate available classes</td>
</tr>
<tr>
<td>Choose class</td>
<td>Validate schedule fit</td>
</tr>
<tr>
<td></td>
<td>Calculates fees</td>
</tr>
<tr>
<td></td>
<td>Request confirmation</td>
</tr>
<tr>
<td>Confirm enrollment</td>
<td>Enroll student in class</td>
</tr>
<tr>
<td></td>
<td>Add fees to student bill</td>
</tr>
<tr>
<td></td>
<td>Provide confirmation of enrollment</td>
</tr>
</tbody>
</table>

Use Case: Login

Properties
- Primary Actor: **Customer**
- Goal: **Customer** logs into the program.
- Level: **Sub-transaction**

Main Success Scenario

1. **Primary Actor** indicates that he/she wishes to log in to the system.
2. **Primary Actor** performs the following in arbitrary order:
   2.1 **The Primary Actor** provides the username.
   2.2 **The Primary Actor** provides the password.
3. **Primary Actor** confirms the provided data.
4. System authenticates the **Primary Actor**.
5. System informs the **Primary Actor** that the Login was successful.
6. System grants access to the **Primary Actor** based on his/her access levels.

Extensions

4a. The provided username or password is/are invalid:
   4a1. System informs the **Primary Actor** that the provided username or password is/are invalid
   4a2. System denies access to the **Primary Actor**.
## 2.3.6. State diagram

- A diagram to describe the behavior of a system
- Require that the system described is composed of a finite number of states

![State Transition Diagram for One Session]

## 2.4. Prototyping

- How do we express early design ideas?
  - Minimal development effort at this stage
- Designs need to be visualized
  - to help designers clarify their own ideas
  - to enable users to evaluate them
- The medium needs to be appropriate for
  - the stage of the process
  - the audience
  - the resources available and the questions that the prototype is helping to answer.
- In HCI designers create prototypes of increasing complexity

## 2.4. Prototyping Dimensions

- **Representation**
  - How is the design depicted or represented?
  - Can be just textual description or can be visuals and diagrams
- **Scope**
  - Is it just the interface (mock-up) or does it include some computational component?
- **Executability**
  - Can the prototype be “run”?
- **Maturation**
  - Revolutionary – Throw out old one
  - Evolutionary – Incorporate design changes
  - Incremental – Modular development

## 2.4.1 Low-fidelity prototypes

- **Paper-based prototypes**
  - a paper mock-up of the interface look, feel, functionality at the high level
  - “quick and cheap” to prepare and modify
  - For early feedback on conceptual design ideas
- **Issues**
  - Robustness – handled by many people
  - Scope – focus on high level only
  - Instructions – designer intervention when users evaluate it
  - Flexibility – can users redesign it ‘on-the-fly’?

## 2.4.1 Low-fidelity prototypes

1. **Sketches**
   - drawing of the outward appearance of the intended system
   - hard to envision a dialog’s
2. **Storyboarding**
   - a series of key frames
     - originally from film; used to get the idea of a scene
     - snapshots of the interface at particular points in the interaction
     - users can evaluate quickly the direction the interface is heading
2.4.1 Low-fidelity prototypes

2.4.1 Storyboard of a computer based telephone

2.4.1 Lo-fi: PICTIVE

► Plastic Interface for Collaborative Technology Initiatives through Video Exploration
► To empower users to act as full participants in design
► Materials
  ▪ Colored pencils, highlighters, pens, sticky notes, icons, menu
  ▪ (plastic) design objects for screen and window layouts
► Equipment
  ▪ shared design surface, e.g. table
  ▪ video recording equipment

2.4.1 PICTIVE Session

► Before a PICTIVE session
  ▪ Users (or developers) generate scenarios of use
  ▪ Developers produce design elements for the design session
► PICTIVE session
  ▪ Introduction: Stakeholders all introduce themselves
  ▪ Tutorials: about areas represented in the session (optional)
  ▪ Brainstorming: ideas for the design
  ▪ Walkthrough of the design: summary of decisions made
► Developers’ roles
  ▪ Act as the computer
  ▪ Take notes, especially those that are not captured by the video camera
  ▪ Get users to go through the design again at the end

2.4.2 Prototyping: Wizard of Oz

► A method of testing a system that does not exist
► Human simulates the system’s intelligence and interacts with user
► Uses real or mock interface
  ▪ “Pay no attention to the man behind the curtain!”
► User uses computer as expected
► “Wizard” (sometimes hidden):
  ▪ interprets subjects input according to an algorithm
  ▪ has computer/screen behave in appropriate manner
► Good for:
  ▪ adding simulated and complex vertical functionality
  ▪ testing futuristic ideas ➞ think about real implementation, though
2.4.3 High fidelity prototypes

► Prototyping with a computer
  ▪ simulate or animate some but not all features of the intended system to engage end users

► Purpose
  ▪ provides a sophisticated but limited scenario to the user to try out
  ▪ provides a development path (from crude screens to functional system)
  ▪ can test more subtle design issues

► Danger
  ▪ user’s reactions are about small things
  ▪ users reluctant to challenge / change the design

2.4.4 Limiting prototype functionality

► vertical prototypes
  ▪ includes in-depth functionality for only a few selected features
  ▪ common design ideas can be tested in depth

► horizontal prototypes
  ▪ the entire surface interface with no underlying functionality
  ▪ a simulation; no real work can be performed