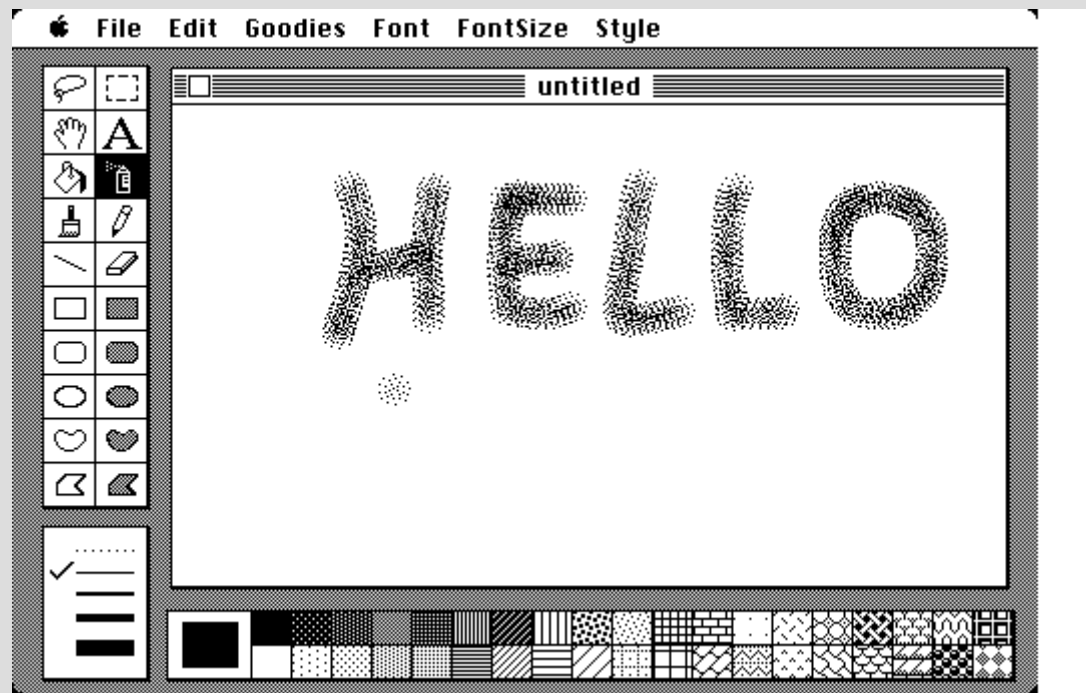


# Graphical Design



# Graphic Design & User Interfaces

- *Information-oriented, systematic graphic design* is the use of typography, symbols, color and other static and dynamic graphics to convey facts, concepts and emotions
- These techniques can be used in user interface design to effectively convey information about system functionality and other data.

# Visible Language

- Visible Language
  - Layout – formats, proportions and grids
  - Typography – typefaces and typesetting
  - Color and texture
  - Imagery – signs, icons and symbols
  - Animation and sequencing – dynamic displays, order of operations
  - Sound

# Marcus' 3 principles of User Interface Design

- 3 basic principles can be applied to the design of menus, control panels, dialogue windows, icons etc:
- Organize
  - Provide user with clear and consistent conceptual structure
- Economize
  - Maximize the effectiveness of a minimal set of cues
- Communicate
  - Match presentation to capabilities of user

# Principle 1: Organize

- Consistency
  - Internal consistency
    - Same conventions and rules for all elements of GUI
  - External consistency
    - Existing platform and cultural conventions across GUIs
  - World consistency
    - Make conventions consistent with real world experience



# Principle 2: Economize

- Make do with what you have ! How many controls does a device really need ?
- Simplicity
- Clarity
  - Design all components so that their meaning is not ambiguous
  - Minimize clutter
- Distinctiveness:
  - Distinguish important properties of essential elements and make the most important elements obvious

# Principle 3: Communicate

- Legibility, Readability, Typography, Symbolism, Multiple Views, Color/Texture
- Legibility
  - Design individual characters, symbols and graphic elements to be easily noticeable and distinguishable
- Readability
  - The display should be comprehensive, inviting and attractive

# Typography

- Use small number of typefaces of suitable legibility (max 3 typefaces, 3 sizes)
- Lines of text should have from 40 to 60 characters
- Appropriate text formats (justified left or centered)
- Avoid abuse of capital letters



# Communicate

- Symbolism
  - Signs, icons, images
  - Icon design is very hard
  - If the icon is not very familiar, then label it
  - Avoid gratuitous images (especially animated ones)
- Color/texture
  - Use appropriate highlighting and (de)emphasis techniques to convey meaningful semantic distinctions
  - Any examples ?

# What is a requirement ?

- Statement about intended product that specifies what it should do and how it should perform
- Specific, unambiguous, clear
  - Time to download any page is less than 5 seconds on a DSL line of 20Mbits/second
  - The playback bar should always indicate the position in the audio stream
- Too vague
  - Appealing to teenage girls (too vague)
  - The buttons should be engaging

# Different kinds of requirements

- Functional
  - What the system should do
  - Historically the main focus of requirements activities
  - Examples: word processor must support variety of formatting styles
- Non-functional
  - Run on a variety of platforms
  - Function on 64M of RAM
  - Delivered in 6 months
- Data
  - What kinds of data need to be stored ? Database ?

# Different kinds of requirements

- Environment or context of use
  - Physical
    - Dusty ? Noisy ? Vibrations ? Light ? Heat ? Humidity ?
  - Social
    - Sharing of files, displays, privacy, locking
  - Organizational
    - Hierarchy, IT departments, user support, communications structure, availability of training
  - Users
    - Novice : step-by-step, constrained, clear information
    - Expert : flexibility, access, power
    - Frequent: short cuts
    - Casual/infrequent: clear instructions, menu paths

# Different requirements

- Usability
  - Learnability, flexibility, throughput,
- IMPORTANT
  - User requirements and usability requirements refer to different things
- Kinds of requirements
  - What factors (environmental, user, usability) ?
    - Self-service cafeteria - paying using credit system
    - Nuclear plant control room
    - Distributed car design teams

# An example

- **Self-service cafeteria**
- *Functional*: Calculate total cost of purchases
- *Data*: Access to price of products in cafeteria
- *Environmental*: Noisy and busy environment, users maybe talking while using the system
- *User*: Majority of users under 25, comfortable dealing with technology
- *Usability*: Simple for new users, memorable for frequent users, efficient, deal easily with user errors

# Data gathering techniques (1)

- Questionnaires
  - A series of questions designed to elicit specific information
  - Questions may require different kinds of answers
    - Yes/No, multiple choice, general comments
  - Often used in conjunction with other techniques
  - Can give quantitative and qualitative data
  - Good for answering specific questions from a large, dispersed group of people

# Data gathering (2)

- Interviews
  - Forum for talking to people
  - Structured, unstructured or semi-structured
  - Pros, e.g. Sample scenarios of use, prototypes, can be used in interviews
  - Good for exploring issues
  - Time consuming, infeasible for large user or dispersed user populations



# Data gathering (3)

- Workshop or focus groups
  - Group interviews
  - Good at gaining a consensus view and/or highlighting areas of conflict

# Data gathering (4)

- Naturalistic observation
  - Spend time with stakeholders in their day-to-day tasks, observing work as it happens
  - Gain insights into stakeholder's tasks
  - Good for understanding the context and nature of tasks
  - But, it requires time and commitment from a member of the design team, and it can result in a huge amount of data
  - Ethnography is one form

# Data gathering (5)

- Studying documentation
  - Procedures and rules are often written in manuals
  - Good source of data about the steps involved in an activity, and any regulations governing a task
  - Not to be used in isolation
  - Good for understanding legislation, and getting background information
  - No stakeholder time, which is a limiting factor in the other techniques

# Data gathering (6)

- Data logging
- Instrument software to record user's activity in a log that can be examined later
- RECORD EVERYTHING
- Easy to do
- Requires a prototype or existing software

# Choosing between techniques

- Data gathering techniques differ in two main ways
  - Amount of time, level of detail and risk associated with the findings
    - Naturalistic observation (2 days of effort, 3 months of training)
    - Interview (1 day of effort, 1 month of training)
  - Knowledge the analyst requires
- Sequential steps, overlapping subtasks ?
- Complex or simple information ?
- Layman or skilled practitioner ?

# Problems with data gathering (1)

- Identifying and involving stakeholders, managers, developers, customer reps etc.
- Involving stakeholders: workshops, interviews, workplace studies
- “Real” users not managers
  - Traditionally a problem in software engineering , better now

# Problems with data gathering (2)

- Requirements management: version control, ownership
- Communication between parties
  - Within development team
  - With customer/user
  - Between users ... different parts of an organization use different terminology
- Domain knowledge distributed and implicit:
  - Difficult to dig up and understand
  - Knowledge articulation: how do you walk ?
- Availability of key people

# Problems with data gathering (3)

- Political problems with the organization
- Dominance of certain stakeholders
- Economic and business environment changes
- Balancing functional and usability demands



# Some basic guidelines

- Focus on identifying the stakeholders needs
- Involve all the stakeholder groups
- Involve more than one representative from each stakeholder group
- Use a combination of data gathering techniques
- Support process with props
- Run pilot session
- Carefully record data – always better to have more

# Data interpretation and analysis

- Start soon after data gathering section
- Initial interpretation before deeper analysis
- Different approaches emphasize different elements e.g class diagrams for object-oriented systems, entity-relationship diagrams for data intensive systems

# Task descriptions

- Scenarios
  - An informal narrative story, simple, “natural”, personal, not generalizable
- Use cases
  - Assume interaction with the system
  - Assume detailed understanding of the interaction
- Essential use cases
  - Abstract away from the details
  - Doesn't have the same assumptions as use cases

# Scenario for shared calendar

“The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals’ calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people’s calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in.”