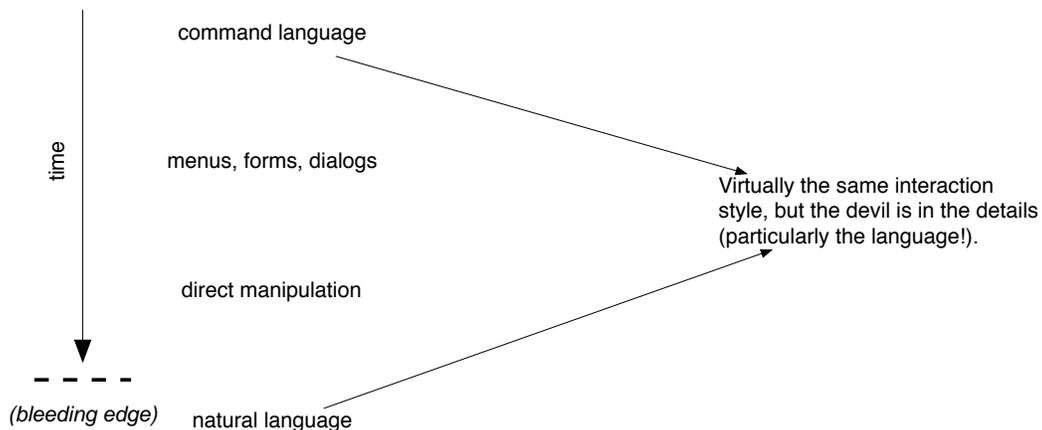# Command and Natural Language Interaction Style

- Many consider language to be the defining human trait: it is the original user interface

- Not surprisingly, it was the first interaction style between humans and computers:

  ◇ Similar in mechanics (but not reality!) to "HHI"

  ◇ Theoretical foundation (grammars, compilers)

  ◇ [Relative] ease of implementation — roughly linear text streams back and forth

# An Interesting Bookend to Interface Design

command language

menus, forms, dialogs

time

direct manipulation

- - - -

*(bleeding edge)*   natural language

Virtually the same interaction style, but the devil is in the details (particularly the language!).

# Design Issues Remain the Same

- Command languages have a reputation of being "hard to use" primarily due to these key properties:

    1. They rely on *recall* rather than *recognition*

    2. Users must *initiate* rather than *respond*

- Nevertheless, some command languages are easier to learn and use than others, and this continues to be for the same reason: the degree of matching between the mental models of the language designer and the user

- Surprise, surprise: OAI helps here too

# Command Languages and OAI

- Task domain (actions, objects) remains the same

- Interface domain: language syntax

    ◇ Task actions and objects must find natural places in the syntax

    ◇ Must strike the right balance between adequate functionality and excessive learning (remember that command languages require *recall* and *initiation*)

    ◇ "Clearly marked exits" and "forgiveness" still hold true: special care with destructive commands; facilities for recording/repeating command history

# Command Language Design Approaches

- Like direct manipulation, command languages benefit from a consistent metaphor: Load/save?  Read/write? Open/close?

  ◇ As usual, watch out for *metaphor shear* — "load message" vs. "load mailbox"; "read message" vs. "read mailbox"; "open message" vs. "open mailbox"

- Simplest approach: one command, one function

  ◇ Best suited to small number of tasks

- Command + argument/option approach

  ◇ Parsing issues: delimiters (white space?  punctuation?); argument order (operating system commands) vs. named arguments (XML)

  ◇ High error rates and low learnability, but mastery may lead to high efficiency and satisfaction (not to mention the ability to accomplish tasks that no menu or direct manipulation interface can accomplish)

- Hierarchical commands: commands as a menu tree

  ◇ Presumes that all commands follow a similar structure

  ◇ Learn the structure once (components and choices), combine in multiple ways

# Command Design Guidelines: Structure

Much studied in the early '80s, so generally supported by empirical data:

- Consistent argument ordering (Barnard et al. 1981)

- Prefer keywords (*replace all* "var" *with* "_var") over symbols (*s/var/_var*) (Ledgard et al. 1980)

- Hierarchical structure and congruence of conceptually related commands (whether similar or opposite) (Carroll 1982)

# Command Design Guidelines: Naming

- Specificity of names: enhances memorability and distinguishability of commands

- Abbreviations: because commands are still typically entered by keyboard, shorter commands are sometimes desirable — but are at odds with specificity and distinguishability

  ◇ Standardized abbreviation rules help here

- Convert commands into menus: display all available commands within a particular context — descriptive output, abbreviated input

# Natural Language Interaction

- The "Star Trek" user interface — still sought after, still generally unsolved



- Natural language in "HHI" carries a major assumption — humans are living in a shared context with a large number of shared assumptions: the world is part of the human-human natural language interface

# The Case Against Natural Language Interaction (NLI)

- Between NLI and direct manipulation styles for the same application — e.g. Ford 1981 NLI checkbook vs. Quicken — direct manipulation has proven more successful despite relative success of NLI

- Frequent, expert users eventually prefer the compactness and precision of more traditional command languages

- "Human-like" dialog has been much-reserached (and encouraged — look up the Loebner Prize), with little tangible progress or application

# Specialized Applications — A Potential Home for NLI

- Natural language querying (NLQ)

  - ◇ SQL is well-known for its English-like syntax while remaining within the bounds of traditional computer languages

  - ◇ Various other products that promote natural language queries: Symantec Q&A, Microsoft English Query, AskJeeves

  - ◇ With the Internet and digital libraries, document searches coming to the fore (a.k.a. information retrieval) — still, keyword vs. natural language?

- Natural language text generation

  - ◇ Natural language as an *output format* — weather, sports transcripts

- Adventure games

  - ◇ The golden age of Infocom — but not around much anymore

- Instructional material

  - ◇ Many games have a "natural language" tutorial section

  - ◇ Other tutorial systems carry a natural language tone

- And take note — speech recognition ≠ NLI!

# Input Devices

Can the keyboard ever be adequately replaced?

- Continuous speech recognition

- Other alternatives…Dasher — http://www.inference.phy.cam.ac.uk/dasher

- Cutting-edge keyboard issues: small devices — foldability, projection

- Handwriting recognition — the visual version of speech recognition

Pointing devices — is the mouse here to stay?

- Direct controls — require interaction with the screen (pens, touch screens)

- Indirect controls — screen is "mapped" to a separate space (mouse, trackball, joystick, trackpoint, touchpad, graphics tablet)

- New approaches — foot controls, eye tracking, 3D tracking, data gloves, Boom Chameleon, haptic feedback, bimanual input, tangible user interfaces, digital paper

Speech and audio — again, the "Star Trek" dream

- Synthesis — accessibility, display-less interfaces

- Recognition — discrete, continuous; training

# Output Devices

- Last few years have seen a drastic shift from CRTs to flat displays

- Latest trends are to either go very large, or go very small — and in both cases, we have real-world analogs (white/blackboards; paper)

  ◇ Increase resolution (maximizing "eyespan")

  ◇ Increase portability (up and coming: e-ink)

- Immersive displays: still out there, though yet to see broad use

- New/recent applications

  ◇ High mobility, ubiquity: cell phones, wearables

  ◇ High bitrate (a.k.a. multimedia): expected everywhere now, in all sizes and devices — H.264

- Non-transient output devices: beyond printers — Braille embossers, "3D printers"

# Collaboration

- The Internet has created a new application area: computer-supported cooperative work (CSCW)

- First envisioned by Englebart

- Decomposition by time and space:

  1. Same time, same place

  2. Same time, different places

  3. Different times, same place

  4. Different times, different places

|  | Same Time | Different Times |
|---|---|---|
| Same Place | "synchronous local"<br><br>control rooms, meeting rooms, desk/wall projections, art/building installations | "asynchronous local"<br><br>equipment logs, team scheduling, group calendars |
| Different Places | "synchronous distributed"<br><br>chat/other messaging; video/ audio conferencing | "asynchronous distributed"<br><br>e-mail, various e-groups, blogs, Wiki |

# It's Still All About This!
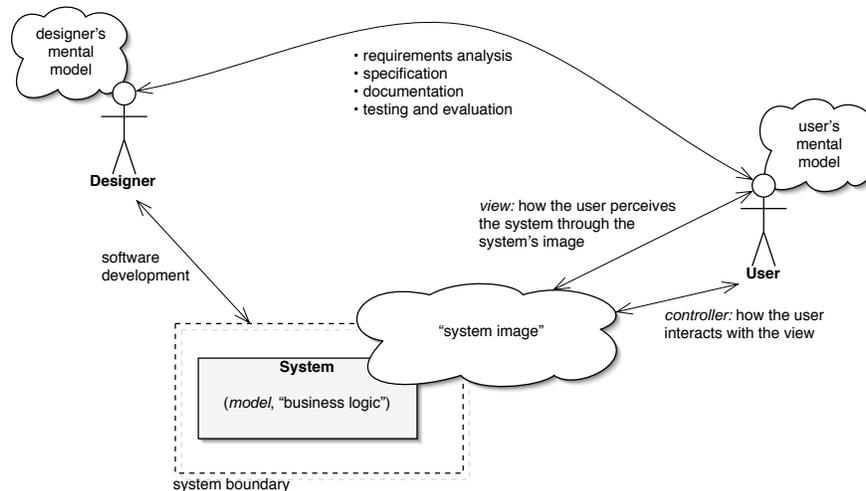


based on Don Norman's model, with some
UML, software engineering, use case modeling,
and MVC tweaks

# Not Covered This Time Around

- Design and evaluation methods and techniques —
  more of Nielsen's focus in *Usability Engineering*

- UI development technology in detail — by choice, we
  focused on Swing for this class

- Specific design issues: quality of service, form vs.
  function, documentation — some in Norman, others
  in Shneiderman

- Information visualization: Edward Tufte tends to be the
  guru here, but there are many others