Lecture 4: Command Languages and Usability

Chapter 8.1

Command Language

• Definition
  – Interactive communication with a computer that requires the user to recall the notation and initiate the action by keyboarding textual elements. Command languages are typically interpreted a single action at a time.

Example Command Language (UNIX)

```bash
> ls -l *.
  foo.dat  slides.prt  exer.prt
> rm foo.dat
> ls -l *.
  slides.prt  exer.prt
```
Elements of Language

- Lexicon
  - Words and punctuation
- Syntax
  - Sequence of words to create a correct sentence
- Semantics
  - "meaning" of a sentence based on the words
- Pragmatics
  - How sentences are used in sequence (discourse)
  - Context
  - Inference

Example Command Language (UNIX)

> ls -l *. *
  foo.dat slides prt exer prt
> rm foo.dat
> ls -l *. *
  slides prt exer prt

Lexical, syntax, semantics, pragmatics……

Usability Questions

- Does the language support necessary functions?
- Is it fast to enter a command?
- Is it easy to recognize what the command might do?
- Is it easy to recall a command?
- Are there few errors when using the language?
UNIX: A case study

- Study done at Bell Labs in 1981-1982
  - Automatic logging of all Unix command transactions in the lab
  - Analyzed
    - Frequency of command usage
    - Transitions between commands (tasks)
    - Error rates of commands

UNIX command usage
(Kraut et al. CHI ’83)

UNIX command transitions
(Kraut et al. CHI ’83)
UNIX
(Kraut et al. CHI ’83)
• 400+ possible commands
• 20 commands (5%) account for 70% of usage
• 14 commands (3.5%) account for 50% usage

UNIX error rates
(Kraut et al. CHI ‘83)
• Types of errors
  – Lexical errors (error in entering command name, abbreviation, parameter specification). Gives message.
  – Syntactic errors (error in expression of a command such as missing parenthesis, wrong order of parameters). Gives message.
  – Semantic errors (valid lexical and syntactic commands but errors where you don’t get what you want). No message.
• Error rates for individual commands ranged from 3% to 50% for expert users!

Usability Problems with UNIX
• Lexicon: Abbreviation not suggestive of function
  – terse
  – inconsistent
  – jargon
• Syntax: Complex syntax
  – Action modifier(s) object(s)
• Semantics: Underutilization of commands
  – Unnecessary complexity to support many functions leads to complexity of most frequent
  – Hard to map commands to tasks
• Pragmatics: Lack of feedback
What this study doesn’t tell us

- How hard it is to learn Unix
  - How much time does it take to get skilled?
  - Different types of users
- How to improve the design

The Basic Goals of Language Design (Chap. 8.1)

- Precision
- Compactness
- Ease in writing and reading
- Speed in learning
- Simplicity to reduce errors
- Ease of retention over time

Higher-Level Goals of Language Design (Chap. 8.1)

- Close correspondence between reality and the notation
- Convenience in carrying out manipulations relevant to user's tasks
- Compatibility with existing notations -- “consistency”
- Flexibility to accommodate novice and expert users
- Expressiveness to encourage creativity
- Visual appeal