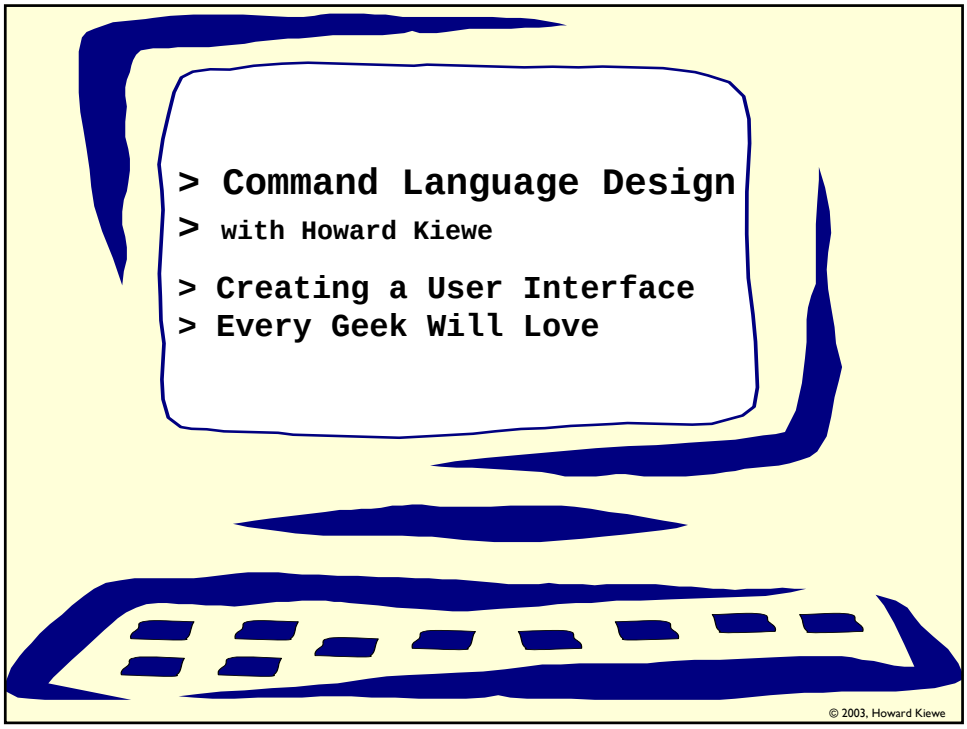


What's a CLI?

- Command Line Interface
- A command language is the part of the CLI with which the user interacts
- An interface used to configure most routers & servers
- $78 billion of router & server hardware is sold each year, compared to $187 billion in PC sales

(Gartner Inc, 2002 a, b, c)

Pros and Cons of CLIs

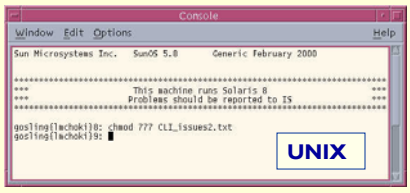
Pros

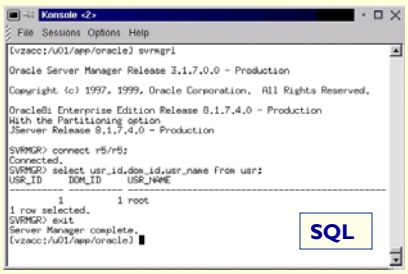
- **Efficient**. Combine commands and don't hunt through the GUI
- **Fast**. Text transmissions consume almost no network bandwidth
- **Flexible**. Write scripts to automate repetitive tasks
- **What many want**. Real geeks don't use GUIs

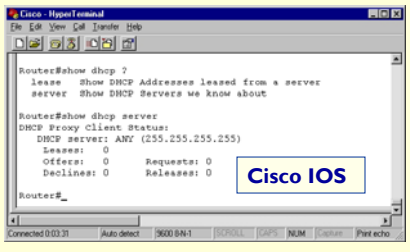
Cons

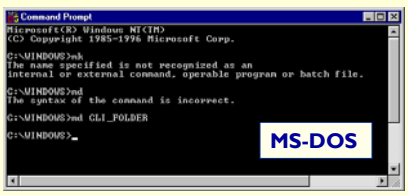
- CLIs rely on recall memory while GUIs rely recognition memory
- Recognition memory is almost 3 times more accurate and almost 2 times faster than recall memory (Nobel, 2001)

CLI Examples

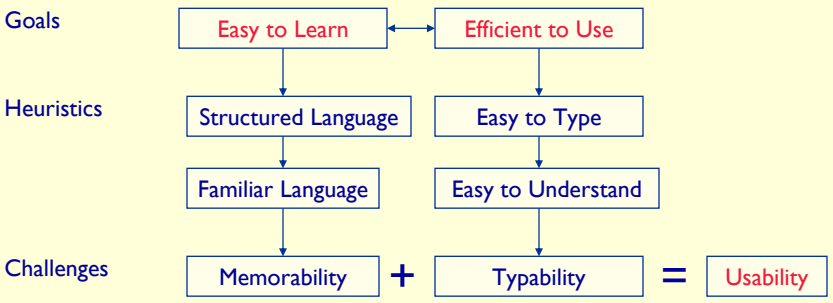








Goals, Heuristics, & Challenges



Heuristics often compete, so you have to find the best trade off

Heuristics: Structure & Familiarity

Structured Language

1. Balance power & simplicity
2. Use a hierarchical structure
3. Define rules & follow them

Familiar Language

4. Model English
   - Use words, not symbols
   - Use familiar punctuation
   - Use parallel words
   - Use the imperative mood
   - If inconsistent, do not model
5. Model the industry standard
6. Use meaningful words
7. Use shorter commands
8. Allow full command words
9. Use distinct words

Heuristics: Typing & Understanding

Easy to Type

10 Abbreviate with truncation

11. Be case-insensitive

12. Avoid shift keys

Easy to Understand

13. Give positive feedback

14. Ask for confirmation

15. Use rich error messages

16. Provide meaningful help

1. Balance Power & Simplicity

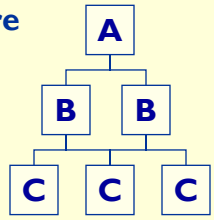
Among UNIX experts, 5% of commands (20/400) accounted for 70% of those used (Kraut, 1983)

- **Power**
  - Many command words
    `add-user`
    `delete-user`
    `edit-user`
  - Example:
    `edit-user larry moe`
  - *Less to type, more to remember*

- **Simplicity**
  - Few command words
    `add-user`
    `delete-user`
  - Example:
    `delete-user larry`
    `add-user moe`
  - *More to type, less to remember*

2. Use a Hierarchical Structure

A recursive tree-like information structure



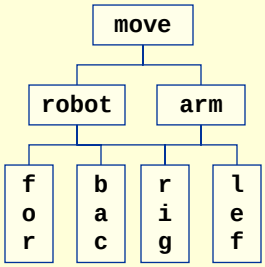
Research:

- Free recall 2–3 times more accurate (Bower, 1969)
- 63% fewer recall & problem-solving errors (Carroll, 1982)

Value:

- Encourages deep levels of processing
- Each level is a mnemonic cue for the next
- Requires fewer command words

2. Use a Hierarchical Structure



✓Do

move robot	forward	
"	backward	
"	right	
"	left	
move arm	forward	
"	backward	
"	right	
"	left	

× Don't

advance
retreat
right
left
raise
lower
swing-out
swing-in

3. Define Rules & Follow Them

Research:

- UNIX error rates 3%–50% by command
- High error rates for non-standard syntax
  (Kraut, 1983)

Example:

- 1–3 command words per command
  Min: `exit` Max: `show ip statistics`
- 0–3 parameters per command
  Min: `exit` Max: `add host host_ip host_name port_num`
- Unique first 2 letters for each command word
  Do: `show / exit` Don't: `show / shutdown`
- Default values for most parameters

4. Model English: Overview

- Use words instead of symbols
- Use familiar punctuation
- Use parallel words
- Use the imperative mood with a verb > object structure
- Do not model English if it's inconsistent

4. Model English: Research

Compared an English-like text editor with a notational editor:

`replace "go" with "gone"` vs. `rs:/go/,/gone/`

- Included novice and experienced users
- 76% more tasks completed with the English-like editor
- 49% fewer errors made
- All users preferred the English-like editor after using it
- Experienced users who preferred a notational editor before the study changed their minds after

(Ledgard, 1980)

4. Model English: Words & Punctuation

	✓Do	×Don't
Use words	`show file` or `s f` `delete file` or `d f`	`ls` `rm`
Avoid symbols	`delete file all` or `d f a`	`rm *`
Use familiar punctuation	`replace 'go' with 'gone'`	`rs:/go/,/gone/`
	`add record`	`add.record;`

4. Model English: Use the Imperative

Tell your computer what to do:

~~Subject >~~ Verb > Object

- **English Language**
  - Go!
  - Help!
  - Add sugar.
  - Find the book.

- **Command Language**

```
exit
help
add route
find file
```

× **Don't**

```
route add
```

4. Model English: Use Parallel Words

- Make related command words semantically & grammatically parallel
- 7 times more errors when nonparallel words were used (Carroll, 1982)

✓**Do**

```
enable ipsec
disable ipsec

insert record
delete record
```

×**Don't**

```
ipsec on
disable ipsec

add record
kill record
```

4. Model English: Unless Inconsistent

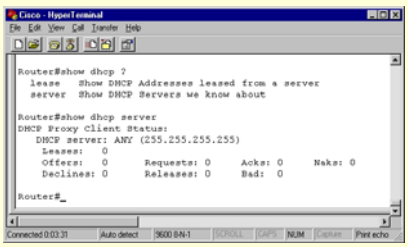
- Task time, number of errors, number of help requests significantly reduced with consistent but non-English syntax
  (Barnard, 1981)

✓Do	×Don't
`search message file`	`search` *for* `file` *in* `message`
`save message ref`	`save message` *as* `ref`
`delete message digit`	`delete digit` *from* `message`

5. Model the Industry Standard

- Balance against other heuristics
- Cisco has 57% of the router market
  (Gartner Inc, 2002 a)
- Cisco IOS is close to an industry standard

- Follows many heuristics
- Doesn't always follow verb > object order



6. Use Meaningful Words

- Meaningful words are easier to remember
- Avoid jargon unless your users think in jargon

✓Do	×Don't
`add`	`concatenate`
`accounting`	`radius`
`user-name`	`network-access-identifier`

7. Use Shorter Commands

- Short-term memory is limited: 7±2 (Miller, 1956)
- Can you pronounce it in under 1.5 seconds? (Schweikcert, 1986)
- Shorter commands are easier to remember
- Shorter commands are faster to type

✓Do	×Don't
`map phone-number`	`add phone-number mapping record`
`add subnet`	`add ip assigned address range`

8. Allow Full Command Words

- When given a choice, 37% of users never abbreviate (Benbasat, 1984)
- If the full word was not known, users forgot twice as many abbreviations (Grudin, 1985)

✓Do	×Don't
`move`	`mv`
`remove`	`rm`
`copy`	`cp`

9. Use Distinct Words

- Words that look similar are difficult to read
- Words that sound similar are easy to confuse
- Words with similar typing patterns are easy to mistype (motor patterns for common words interfere with others)
- Words that start the same are difficult to abbreviate with truncation

✓Do	×Don't
`send`	`send`
`find`	`search`
`choose`	`select`

10. Abbreviate with Truncation

- Having an abbreviation rule improves recall
- Users recall more with a rule than when they make up their own abbreviations
- Simple truncation is the most effective rule, especially for encoding (Ehrenreich, 1985)
- Well-known acronyms can be used instead of words

✓Do		×Don't
`ethernet`	`eth`	`ethrnt` or `ethr`
`show`	`s`	`shw` or `sw`
`delete`	`del`	`dlt`
`ppp`	`ppp`	`point-to-point-protocol`

11. Be Case-Insensitive / 12. Avoid Shift Keys

- Case differences are hard to remember
  `IPSec` or `IPSEC` or `ipsec`? `eMail` or `email` or `e-mail`?
- Shifted characters take longer to type
- Shifted characters are more error prone

✓Do	×Don't
`disable ipsec`	`Disable IPSec`
`enable dynamic-ip`	`enable dynamic_ip`
`find 'user name'`	`find "user name"`

13-14. Make Conversation!

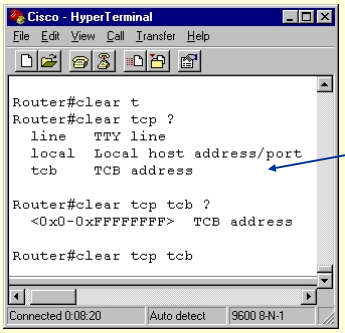
- Give positive feedback
- Ask for confirmation

	✓Do	×Don't
Feedback	> add user jim99 jim99 has been added to the user database >	> add user jim99 >
Confirmation	> delete user jim99 Are you sure you want to delete jim99.[y/n]? >	> delete user jim99 >

15. Use Rich Error Messages

✓Do	×Don't
> add usr jim99 Did you intend to type: add user jim99 [y/n]?	> add usr jim99 Invalid command
> edit timeout You forgot to enter the timeout period. Use a value of 1-60 seconds.	> edit timeout Parameter invalid or missing
> edit timeout 65 The timeout period you entered is out of range. Use a value of 1-60 seconds	> edit timeout 65 Parameter invalid or missing

16. Provide Meaningful Help



Make your 68 characters count:

- Don't just restate the command word
- Hint at what's to come

`TCB address in hex format`

CLI Design Lifecycle

- **Build support**
- Competitive analysis
- User and task analysis
- Card sorting
- Draft, organize, review
- **Build support**
- Iterative testing
- Wizard of Oz method
- Proof read all interface text
- **Build support**

Resources & References

Card Sorting

Software: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/410

Information: http://www.stcsig.org/usability/topics/cardsorting.html

Command Language Design Guidelines

Mayhew, D. J. (1992). *Principles and Guidelines in Software User Interface Design.* Prentice Hall: New Jersey.

References

Barnard, P. J. et al. (1981). Consistency and compatibility in human-computer dialogue. *International Journal of Man-Machine Studies, 15*, 87–134.

Benbasat, I. & Yair, W. (1984). Command abbreviation behavior in human-computer interaction. *Communications of the ACM, 27*(4), 376–383

Bower, G. H., Clark, M. C., Lesgold, A. M., & Winzenz, D. (1969). Hierarchical retrieval schemes in recall of categorized word lists. *Journal of Verbal Learning and Verbal Behavior*, 8, 323–343.

More References

Carroll, J. M. (1982). Learning, using and designing command paradigms, *Human Learning, 1*(1), 31-62.

Ehrenreich, S. L. (1985). Computer abbreviations: Evidence and synthesis. *Human Factors, 27*(2), 143–156.

Gartner Inc. (2002 a). *LAN Equipment and Routers Worldwide, 2000-2001: Market Trends*

Gartner Inc. (2002 b). *Servers Worldwide Quarterly Statistics, 4Q01*

Gartner Inc. (2002 c). *Worldwide PC Forecast Scenarios Update: 2002-2003*

Grudin, J. & Barnard, P. J. (1985). The cognitive demands of learning and representing command names for text-editing. *Human Factors, 26*(4), 407–422.

Kraut, R. E., Hanson S. J. & Farber J. M. (1983). Command use and interface design. *Proceedings, CHI '83,* 120–124.

Ledgard, H. F. et al. (1980). The natural language of interactive systems. *Communications of the ACM, 23,* 556–563.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97.

Nobel, P. A. & Shiffrin R. M. (2001). Retrieval processes in recognition and cued recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27*(2), 384-413.

Schweikcert, R. & Boruff, B. (1986). Short-term memory capacity: Magic number or magic spell? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12*, 419-425.

Contact Information

Howard Kiewe
Usability & Documentation Consultant
hkiewe@videotron.ca
(514) 485-6373
(514) 345-7900 x 5566

STC Montréal
stc-montreal.org

UPA Montréal
upa-mtl.org