Informatics 134

Project in User Interaction Software

Software User Interfaces

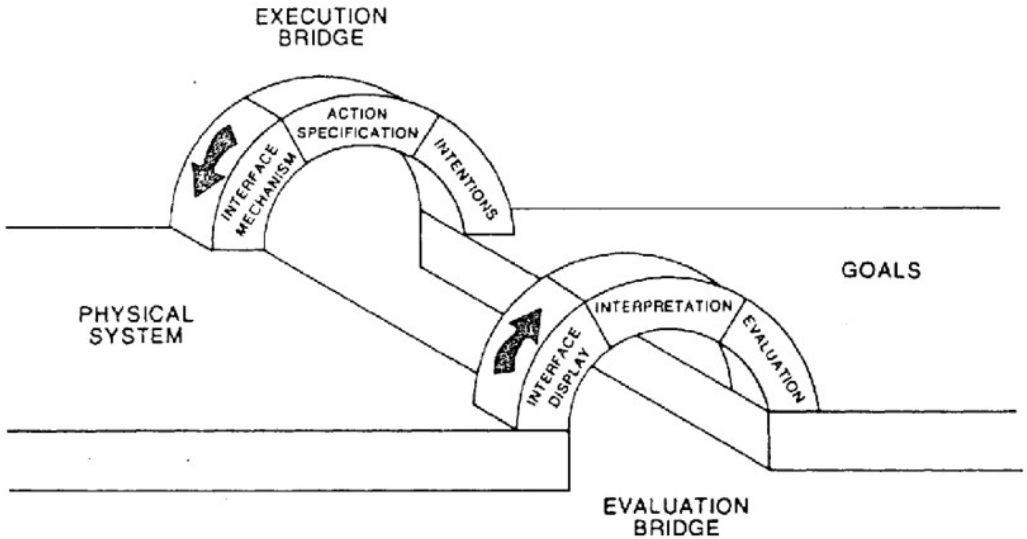
Let’s revisit the gulfs…

The gulf of execution:

The distance between what a user perceives and what a system supports

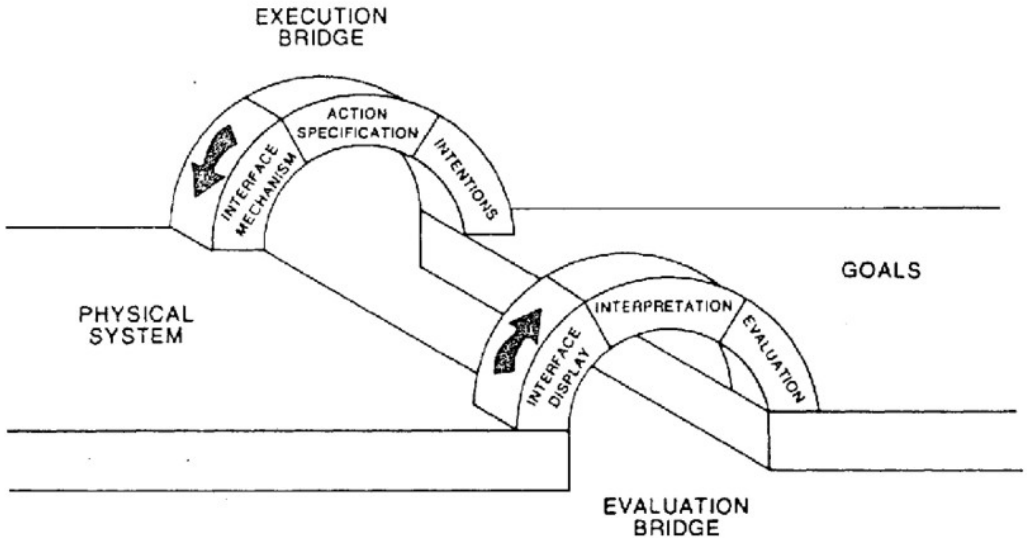
The gulf of evaluation:

The distance between how a user assesses system state and how well the system supports discovery and interpretation of that state



Software User Interfaces

As software designers and developers, your goal is to narrow the gulfs.



Software User Interfaces

So let’s talk about how

There is a amazing body of work that we can learn from...

Software User Interfaces

10 Usability Heuristics for User Interface Design

Summary: Jakob Nielsen's 10 general principles for interaction design. They are called "heuristics" because they are broad rules of thumb and not specific usability guidelines.

By Jakob Nielsen on Apr. 24, 1994; Updated Nov. 15, 2020

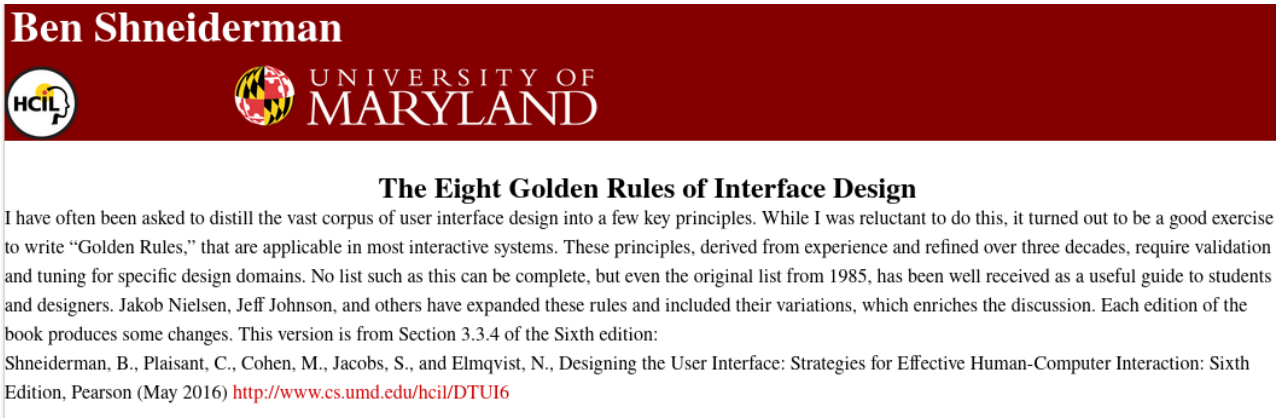
Topics: Heuristic Evaluation, Human Computer Interaction, Web Usability

Share this article:

https://www.nngroup.com/articles/ten-usability-heuristics/

Jakob Nielsen's general principles for interaction design...

Software User Interfaces


Ben Shneiderman

HCIL

UNIVERSITY OF MARYLAND

The Eight Golden Rules of Interface Design

I have often been asked to distill the vast corpus of user interface design into a few key principles. While I was reluctant to do this, it turned out to be a good exercise to write "Golden Rules," that are applicable in most interactive systems. These principles, derived from experience and refined over three decades, require validation and tuning for specific design domains. No list such as this can be complete, but even the original list from 1985, has been well received as a useful guide to students and designers. Jakob Nielsen, Jeff Johnson, and others have expanded these rules and included their variations, which enriches the discussion. Each edition of the book produces some changes. This version is from Section 3.3.4 of the Sixth edition:

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., and Elmqvist, N., Designing the User Interface: Strategies for Effective Human-Computer Interaction: Sixth Edition, Pearson (May 2016) http://www.cs.umd.edu/hcil/DTUI6

Ben Shneiderman's Eight Golden Rules of Interface Design

https://www.cs.umd.edu/users/ben/goldenrules.html

Software User Interfaces

Bruce Tognazzini's Principles of Interaction Design...

Home | Interaction Design Section | Living Section | About Bruce Tognazzi

First Principles of Interaction Design (Revised & Expanded)

5 Mar 2014 in First Principles, HCI Design, Human Computer Interaction (HCI), Principles of HCI Design, Usability Testing

https://asktog.com/atc/principles-of-interaction-design/

Software User Interfaces

There are many more, but these are the gold standards.

We can distill them down to a few basic principles to follow when we start designing user interfaces.

1. **Consistency.**
2. **Control.**
3. **Comfort.**
4. **Cognitive load.**

Software User Interfaces

Consistency

- Visual artifacts follow a theme or pattern

- Common behaviors remain the same

- Support user expectations

Software User Interfaces

Control

- Keep users aware of system state

- Support the transition from novice to expert

- Provide feedback and support error recovery

Software User Interfaces

Comfort

- Avoid clutter, unnecessary UI elements

- Avoid repetition

- Apply clear, understandable terms

- Take advantage of existing metaphors

Software User Interfaces

Cognitive Load

- Strive to reduce cognitive burden

- Support recognition over recall

- “form follows function” (make things work as users expect)

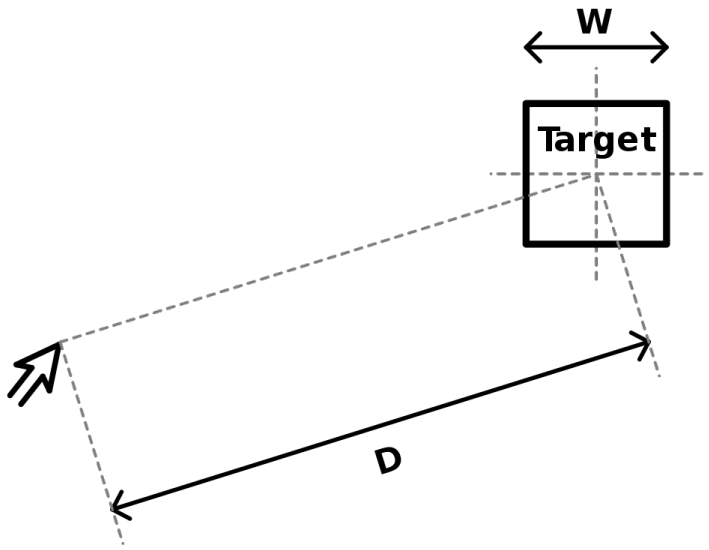
- Avoid overstimulating visual layouts

Software User Interfaces

A few considerations...

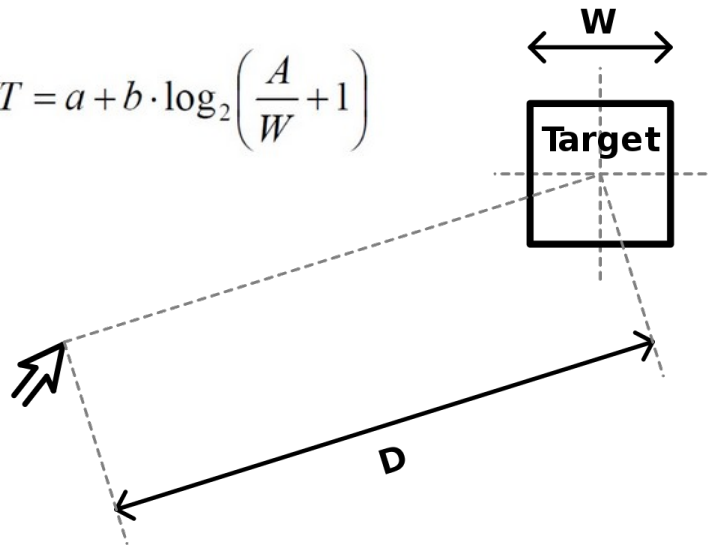
Look familiar?

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$



Fitts' law

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$



MT - the time it takes to move a pointer to a target

A - the distance of movement

W - the size of the target

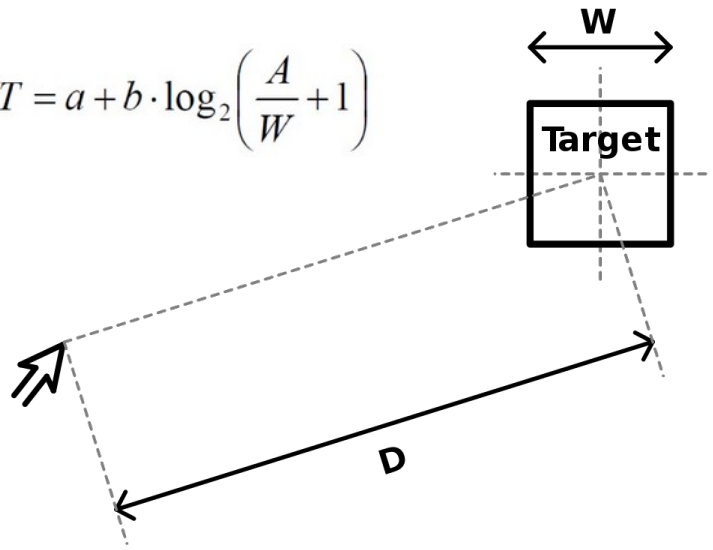
a - device specific coefficient (e.g., time for brain to trigger hand to move mouse)

b - coefficient for efficiency of movement (e.g., sticky mouse)

Software User Interfaces

Fitts' law

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$



How do we use it?

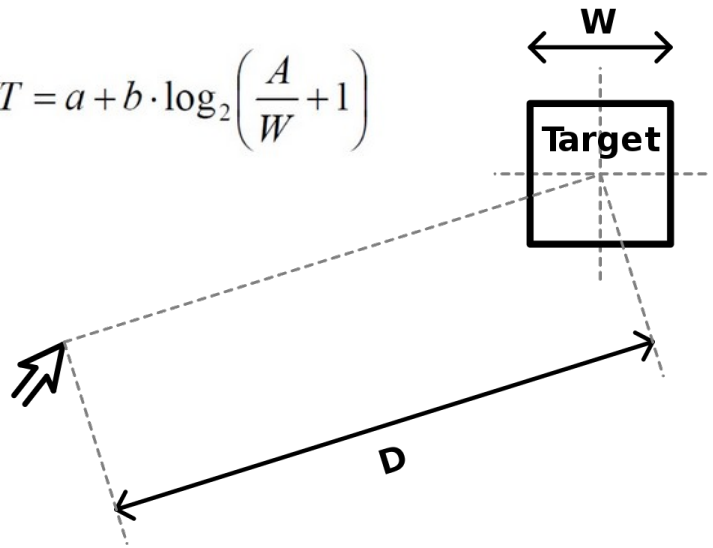
Fitts' law tells us that the larger the target (W), the faster we will be able to accurately interact with that target.

We can apply this basic principle to user interface design.

Software User Interfaces

Fitts' law

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$



How do we use it?

The way a user points directly influences the effectiveness of an interface. So…

We can leverage these rules to improve our design:

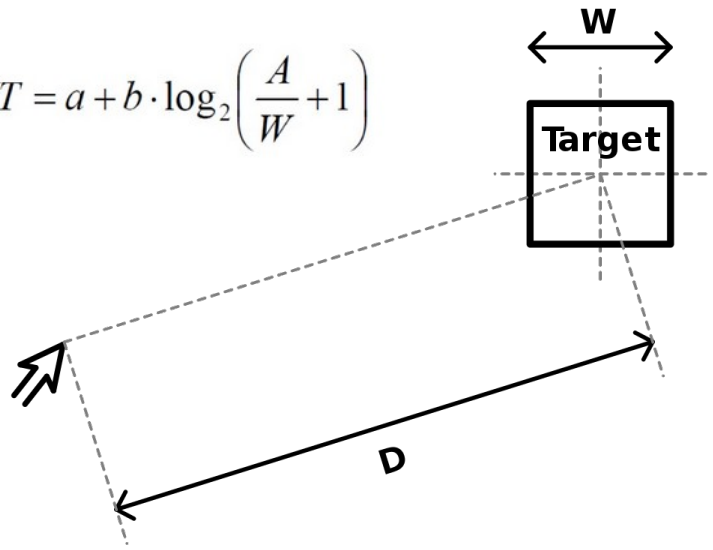
- Constraint of a screen
- Distance between common widgets

Software User Interfaces

Fitts' law

Demo

$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$



Software User Interfaces

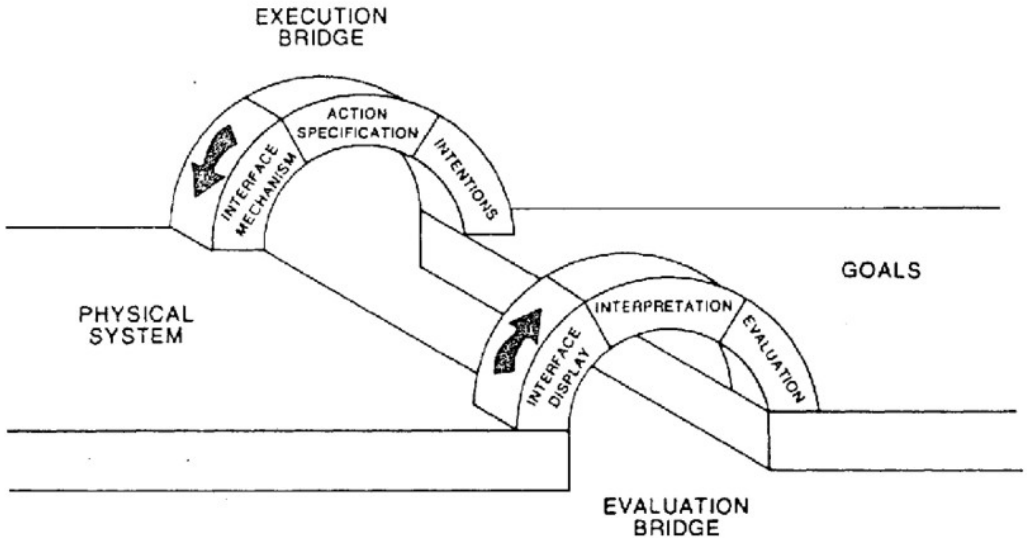
Narrowing the Gulfs:

1. Consistency.

2. Control.

3. Comfort.

4. Cognitive load.



Software User Interfaces

A few considerations…

Pointing efficiency and Fitts' Law

Learnability

Software User Interfaces

Learnability

How can interfaces bridge the gulfs by helping users learn?

Software User Interfaces

Learnability

How can interfaces bridge the gulfs by helping users learn?

- Provide awareness of functionality

- Help locate functionality

- Inform use of functionality

Software User Interfaces

Learnability

During evaluation if you encounter questions like:

- What is this?

- What does this do?

- What does this mean?

- How do I return to X?

A good indication that your interface should provide help.

Software User Interfaces

Strategies to improve learnability

Natural language?

- Labels

- Title bars/headers

What words would you use to describe a feature that posted a message to some social media service?

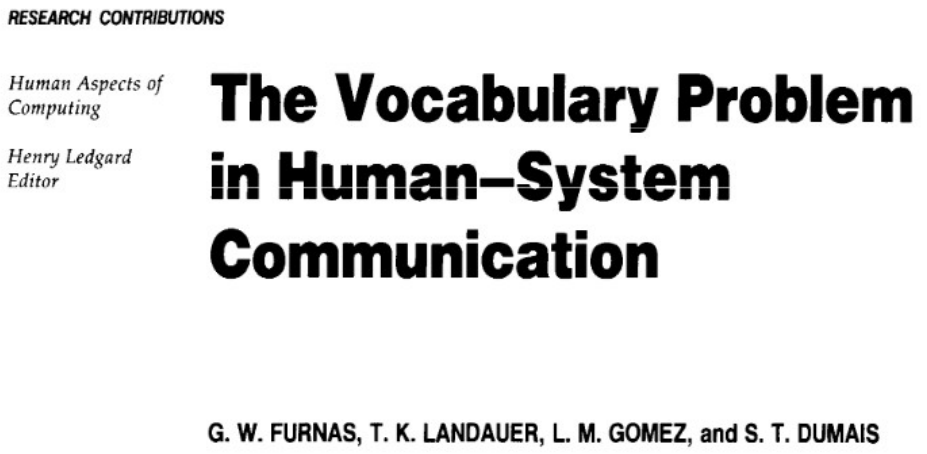
Software User Interfaces

Strategies to improve learnability

To achieve 90% phrase recognition, systems must use roughly 20 unique phrases

So…

low recall vs. low precision



RESEARCH CONTRIBUTIONS

Human Aspects of Computing

Henry Ledgard Editor

The Vocabulary Problem in Human–System Communication

G. W. FURNAS, T. K. LANDAUER, L. M. GOMEZ, and S. T. DUMAIS

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The vocabulary problem in human-system communication. Commun. ACM 30, 11 (Nov. 1987), 964–971. DOI:https://doi.org/10.1145/32206.32212

Software User Interfaces

Strategies to improve learnability

If one or two words are insufficient for learning, what else can be done?

Software User Interfaces

Strategies to improve learnability

Tutorials:

- Teach it!

- But, have you ever read the help docs for a program?

Software User Interfaces

Strategies to improve learnability

The Paradox of the Active User

- Most users need documentation to learn interfaces

- Most users find learning a distraction to their immediate goals

Paradox of the Active User[1]

John M. Carroll and
Mary Beth Rosson

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. In Interfacing thought: Cognitive aspects of human-computer interaction (pp. 80-111).

Software User Interfaces

Strategies to improve learnability

Integrated Search:

-

Software User Interfaces

Strategies to improve learnability

Integrated Search:

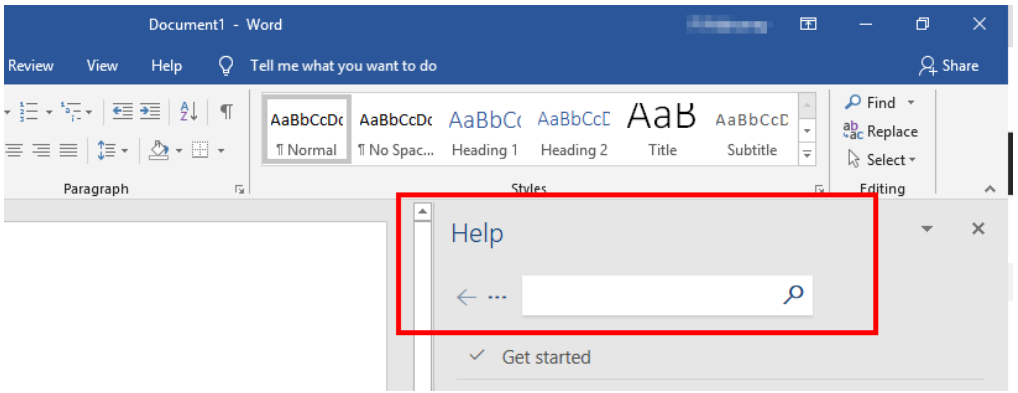
- Vocabulary problem

- Negative perceptions on accuracy

- Disconnect between practical use and documentation

- Answer not available!

- Removes user from task at hand



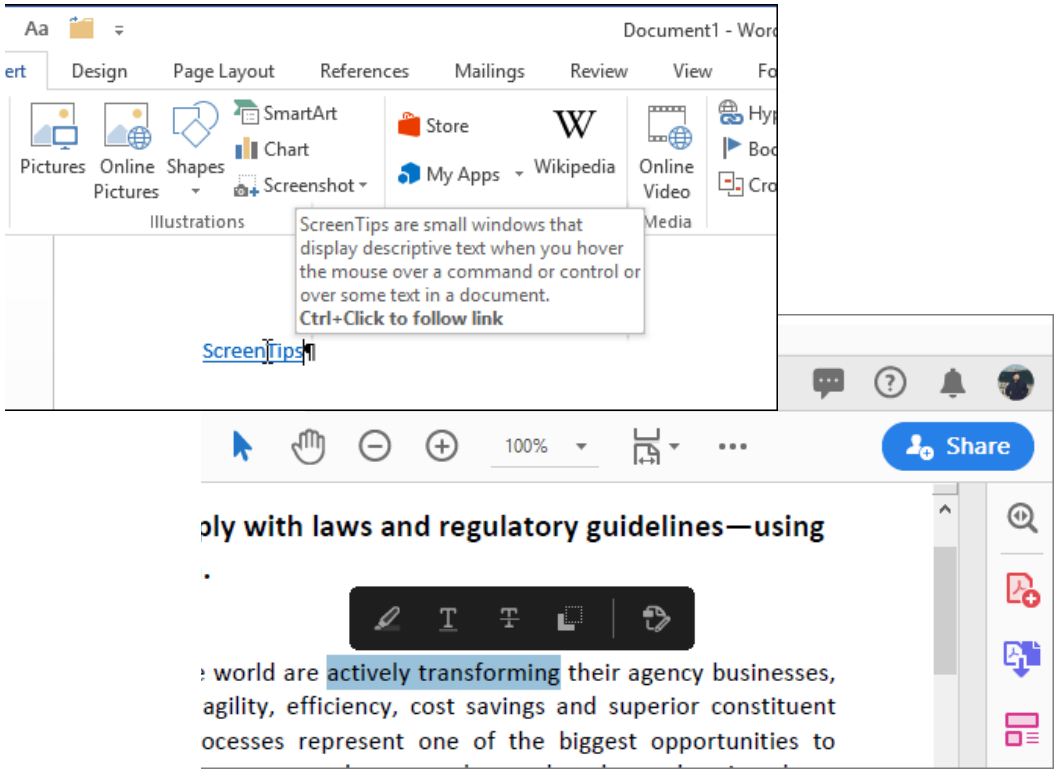
Software User Interfaces

Strategies to improve learnability

Alternatives to search:

- Ballon Help/Tooltips/ScreenTips

- Good for simple tasks, less helpful for multi-step tasks



Software User Interfaces

Strategies to improve learnability

Seek vs. Infer:

- Rules-based, if x then y

- Assume novice, always display help

- Monitor behavior to model when help is needed...

Software User Interfaces

Strategies to improve learnability

Seek vs. Infer

Rather than rely on the user to know, infer based on user behavior.

Principles of Mixed-Initiative User Interfaces

Eric Horvitz
Microsoft Research
Redmond, WA 98025 USA
+1 425 936 2127
horvitz@microsoft.com

Horvitz, E. (1999, May). Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 159-166).

Software User Interfaces

Strategies to improve learnability

Horvitz's Lumiere system attempted to infer user goals from a history of their actions.

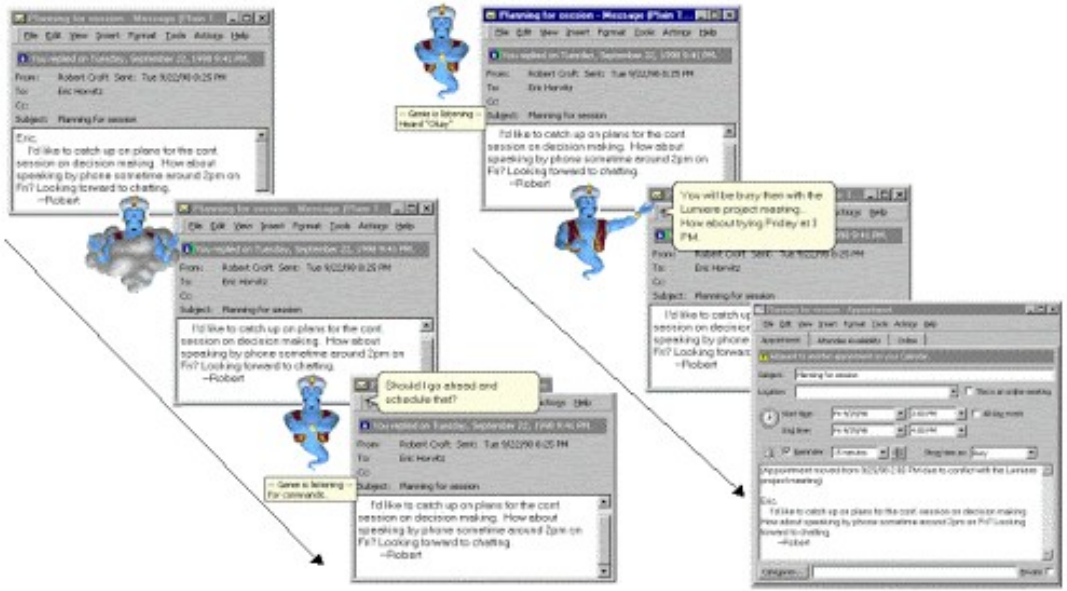


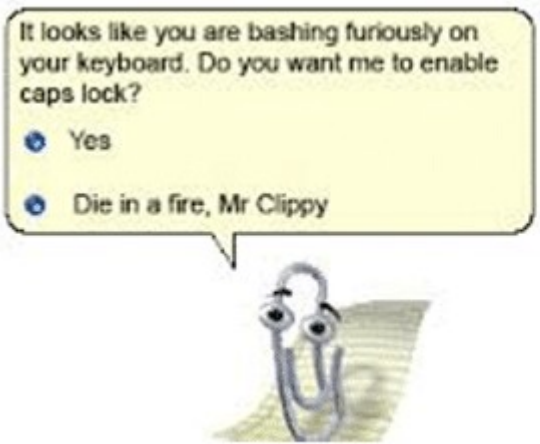
Figure: Lookout operating in its social user-interface modality.

Software User Interfaces

Strategies to improve learnability

The final implementation might look familiar…

Opted for a rules-based approach rather then action-history modeling.



Software User Interfaces

Strategies to improve learnability

What are some actions that we can use to infer the need for help?

Software User Interfaces

Strategies to improve learnability

What are some actions that we can use to infer the need for help?

- Pause in action?

- Undo or erase?

- Disruption to expected flow?

- Avoidance or lack of use?

Software User Interfaces

Strategies to improve learnability

Can we adjust level of help based on experience?

- Novice to expert…

- Ask? Profile setup…

Software User Interfaces

Strategies to improve learnability

Can we adjust level of help based on experience?

- Infer based on behavior?

- Hurst, et al., implemented a classifier that could identify skill based on accuracy of interaction.

CHI 2007 Proceedings • Expert/Novice April 28-May 3, 2007 • San Jose, CA, USA

Dynamic Detection of Novice vs. Skilled Use Without a Task Model

Amy Hurst, Scott E. Hudson, Jennifer Mankoff
Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{akhurst,scott.hudson,jmankoff}@cs.cmu.edu

Hurst, A., Hudson, S. E., & Mankoff, J. (2007, April). Dynamic detection of novice vs. skilled use without a task model. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 271-280).

Software User Interfaces

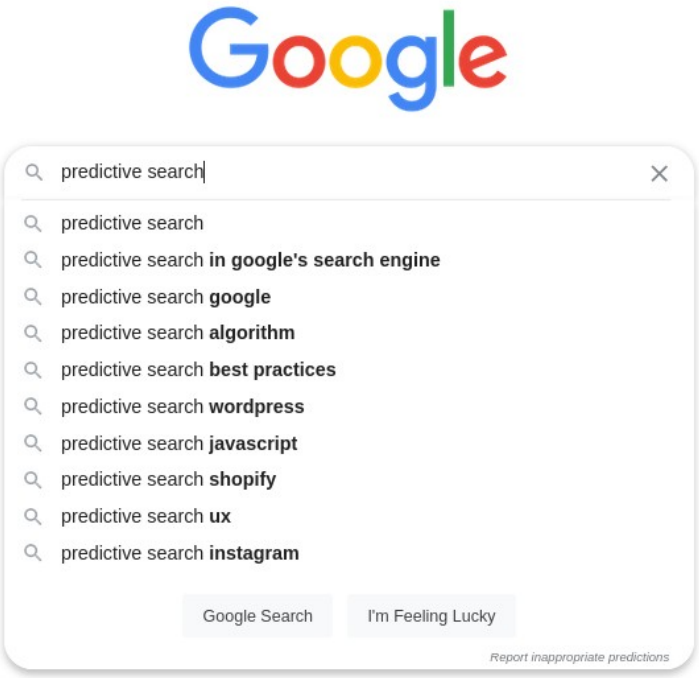
Strategies to improve learnability

To the crowd?

- Always on, changes how we build models.

- If one person needs help, it's like others will too.

- Trade-offs?



Retrospective Think Aloud

Retrospective Think Aloud

Some advantages…

- Avoids concurrent in-task verbalization

- Reduces risk of researcher influence

- Retains focus on task-at-hand, more closely simulating real world conditions

- Enables researcher to collect other usability data (performance, time, errors)

Retrospective Think Aloud

Some disadvantages…

- Time cost: Typically longer than traditional concurrent think aloud study

- Some studies suggest retrospective data is not as strong as *in situ* reflection

- Some added complexity in study setup (recording playback)

Retrospective Think Aloud

Why does RTA work for us?

- Balances researcher role in study, better for learning (I think)

- Gain familiarity with different types of data collection in one study.

- TA's in general are considered the most valuable method for developing good usability.

Retrospective Think Aloud

How is it done?

- A participant is assigned a task or series of tasks.

- The research moderator brief overview of the system to be evaluated.

- The participant is asked to complete all tasks.

- The research team observes, takes notes, and tracks relevant usability data.

- The participant is asked to watch a recording of their actions, comment on what they were thinking, and answer questions from the moderator.

Retrospective Think Aloud

“The participant is asked to watch a recording of their actions, comment on what they were thinking, and answer questions from the moderator.”

For your RTA, you will make some slight adjustments...

Retrospective Think Aloud

The INF 134 RTA

- You do not have to play back your recording.

- You are all moderators.

- You will ask the participant to reflect on specific portions of the task in response to your observations.

- You will complete the study by asking the participant to discuss any likes, dislikes, or other thoughts on their experience with your prototype.

Retrospective Think Aloud

The INF 134 RTA

- Do your best to remain silent during task completion.

(Not always possible, but try.)

- Look for points of confusion, incorrect decisions, breakdowns, and errors. Note them and discuss during reflection.