A Heuristic Evaluation of a World Wide Web Prototype

Michael D. Levi and Frederick G. Conrad

Background

The Bureau of Labor Statistics (BLS) is the principal fact-finding agency for the Federal Government in the broad field of labor economics and statistics. It has a dual role as the statistical arm of the Department of Labor and as an independent national statistical agency that collects, processes, analyzes, and disseminates sensitive economic and statistical data to the American public, Congress, other Federal agencies, State and local governments, business, and labor. BLS conducts surveys of consumer prices, employment, unemployment, wages, benefits, producer prices; international comparisons; productivity and costs; etc. The Bureau of Labor Statistics has one primary product -- economic data. Making that data easily accessible in formats convenient for its users is central to the mission of the agency.

In January, 1994, BLS began providing the public direct access to its central data repository via FTP and gopher over the Internet. This service has become quite popular and continues to grow as more data users learn of its availability. Nonetheless, users consistently request significant functional upgrades to the services provided.

The Prototype

By the summer of 1994, interest in the World Wide Web (WWW) had begun to grow within the Bureau. In order to investigate this new information dissemination channel, a group of developers began creating a prototype WWW system. The developers were primarily systems analysts, with some representation by the BLS Office of Publications. In three months time this group finished a prototype public access system containing just over 100 HTML pages and three embedded mini-applications: a "comments and suggestions" form, a text search through abstracts of economic and statistical research papers, and a forms-based interface to the Consumer Price Index. When the prototype was demonstrated to senior management it was greeted with enthusiasm, and development of a comprehensive, production-level system was approved.

The prototype itself was not designed through accepted software engineering methods. Instead, it was treated as a collection of electronically published documents that grew without careful planning. None of the developers was familiar with WWW technology, and a large part of prototype creation was spent discovering the capabilities and limitations of the Web. During weekly meetings of the development group (this was a part-time activity for all concerned) an overall design was established, and a set of stylistic agreements evolved. One group member took on the overall coordination of pages and subsystems; at the end of the page creation process she reviewed and modified the pages for consistency.

As planning for a production system began, it became clear that a much more rigorous development process would be required. The decision was made to treat the collection of Web pages as a software system, and to follow an established software development life

cycle, beginning with requirements gathering, moving through analysis and design, implementation, testing, and, finally, distribution. The prototype was seen as a learning experience, to be mined for reusable components but discarded as an independent system.

One of the key components of the prototype was its user interface. The full-fledged WWW system will likely become one of the major points of contact between the Bureau of Labor Statistics and its user base. For many users this system will be the only grounds on which they can judge BLS. Thousands of users will be obtaining mission-critical data from this source. Ease of learning, ease of use, and general user satisfaction, along with quality and comprehensiveness of content and functional capabilities, will determine the success or failure of the effort.

The project leader thus decided to perform a usability test on the prototype, and feed the results into the production system development process. Because the project was limited in resources and time, the evaluation had to be quick and cheap. At the same time, any evaluation would need to reflect the enormous diversity of experience and perspective found in the user community connected over the Web.

The Method

Usability inspection, as defined by Jakob Nielsen, is "the generic name for a set of methods based on having evaluators inspect or examine usability-related aspects of a user interface." [1] Of the eight inspection methods cataloged by Nielsen, heuristic evaluation is the least formal and involves having usability specialists judge whether aspects of a given interface conform to a list of established usability principles, known as the heuristics.

Heuristic evaluation, along with the other inspection methods, differs from more conventional empirical usability testing in significant ways: evaluators are not drawn from the user community, evaluations take less time, evaluations are easier to set up and run, and evaluations cost less. "It is easy (can be taught in a half-day seminar); it is fast (about a day for most evaluations); and it is as cheap as you want it." [1] The promise of ease, speed, and low cost attracted us; we decided to kill two birds with one stone, and both evaluate the prototype and explore the method simultaneously. The objective was to determine the usability shortcomings of the prototype so they would not be repeated in the final product, and to determine whether heuristic evaluation had promise for future projects within BLS and similar organizations.

The evaluation was carried out by two experimenters. In order to prepare for the evaluation session, the experimenters prepared two documents. The first was a project overview, describing the objectives, target audiences, and expected usage patterns of a WWW public access system. The second was a list of usability principles, or heuristics. The heuristics were derived from a general-purpose list given by Nielsen, supplemented by hypertext-specific principles drawn from Shneiderman and Kearsley [2] and the experimenters' intuition. It is important to note that the prototype developers had not been given either of these documents during their development process; the documents were drawn up after the fact.

[Sidebar #1: Project overview]

[Sidebar #2: Usability Principles (Heuristics)]

The next step was to identify the evaluators. Nielsen recommends using 3 - 5 usability experts as evaluators. To broaden the evaluator perspectives, but also to help us evaluate the method, we decided to run the evaluation process in parallel with two separate groups. The first was a group of four user interface (UI) experts drawn from within our organization. These four had participated in a three-year BLS User Interface Working Group. This group met weekly, attended classes and conferences, created user interface guidelines for the Bureau at large, and was instrumental in developing an internal training curriculum. The second evaluation group consisted of four developers who had worked on the prototype.

The evaluation process was almost identical for both groups. The two experimenters met with each group for 45 minutes to explain the purpose of the sessions, preview the process, present the project overview and heuristics, and answer any questions. In addition, the UI experts were given a quick training session on Mosaic (the WWW browser used during the evaluation), which most of them had not previously used. At the end of this initial briefing, individual evaluation sessions were scheduled with each evaluator.

The individual evaluation sessions lasted about one and one half hours each. The evaluators were instructed to browse through the WWW prototype (no specific scenarios were presented, though the four sample usage patterns were emphasized), identifying potential usability problems, and tying each problem found to the specific heuristic it violated. Multiple heuristics could be linked to any given violation. The experimenter manually recorded data from all sessions. For the UI experts, a developer was present to assist with any browser difficulties, and to answer questions as they arose.

After all the individual sessions had been completed, each group met as a whole for about an hour and a half. During these meetings, facilitated by the experimenters, each evaluator presented the violations she/he had found along with the heuristic that was violated, and a composite list was assembled for each group. During this session, suggestions for improvement were also discussed.

After the debriefing session, the experimenters formatted the composite list of violations as a rating form, and sent it via e-mail to each evaluator. Evaluators were requested to assign severity ratings to each violation on a five-point scale. The evaluators' severity ratings were e-mailed back to the experimenters, and the individual lists aggregated and analyzed.

[Sidebar #3: Severity Rating Scale]

In sum, the process took approximately 4 hours for each evaluator, and (since portions of the process took place with all evaluators meeting together) approximately 10 hours per group for the experimenters. This does not count the initial planning or final analysis time. Each group completed its effort in two days.

Results: Quantitative

Surprisingly, this study found relatively few differences between the two evaluator groups.

Each group identified numerous problems in the prototype: the UI experts found 102 violations of the heuristics, the developers found 64. Much of this difference can be attributed to a single UI expert who was especially prolific.

Moreover, when the proportion of total violations attributed to each of the eight heuristics was calculated, those proportions were roughly the same for both groups of evaluators (r = .69)

[Sidebar #4: Proportion Violations Graph]

The biggest discrepancies were found in heuristic #2 (consistency) and heuristic #7 (progressive levels of detail). The UI experts found more consistency violations than did the developers, while the pattern was reversed for progressive levels of detail. One explanation is that developers were well acquainted with the application's intended functionality and so did not find inconsistencies to be particularly disruptive. In the same vein, developers were very familiar with the content of the prototype's pages, and thus were particularly attuned to inappropriate granularity of information. The UI experts, coming to the system without previous exposure, were disconcerted by inconsistencies and didn't have enough time to recognize content irregularities.

Severity rating judgments were also comparable between the two groups. The average rating for the UI experts was 3.06 and for the developers was 2.81. The average ratings grouped by heuristic were correlated between the groups of evaluators (r = .86).

Severity ratings per heuristic, however, varied in size and agreement. The lowest severity ratings and highest variance was encountered with heuristic #5 (aesthetic and minimalist design). The highest severity ratings and lowest variance was encountered with heuristics #7 (progressive levels of detail) and #8 (navigational feedback). Heuristic #6 (chunking) also had high severity ratings and low variance, but the number of observed violations was quite small.

[Sidebar #5: Severity Ratings by Group]

These results indicate to us that aesthetic judgments are quite subjective, leading to disagreement on their importance (relatively high variance) though in general such problems are not deemed crucial. Hierarchy and navigation problems, in contrast, were judged important by all evaluators (relatively low variance) suggesting that these are more objective criteria whose violation should be addressed.

Results: Thematic

In general, both groups identified detailed and specific problems at all levels of the system.. Some violations were deemed out of bounds by the experimenters as they related

to the browser being used rather than the pages being examined, or had to do with content rather than interface issues. It should be noted that no evaluator identified any problems with the overall structure or organization of the system.

If this evaluation had been performed on an early version of a production system, the full list of specific violations detected would have been passed on to the project team for analysis, prioritizing, and correction. This list is the essential deliverable of such an evaluation. In this case, however, the system being evaluated was a throw-away prototype, and the experimenters were looking for broader lessons to pass on to a new development group who would be starting from scratch. Thus the experimenters grouped related problems into larger themes. The overlap in some of the themes reflects the overlap in some of the heuristics.

[Sidebar #6: Thematic Problems Identified]

Though many of the violations identified were the same between the two groups, some differences are apparent. The UI expert group was clearly more attuned to broad usability issues (consistency, visual clutter). Moreover, the UI experts' unfamiliarity with the particulars of this system prevented them from identifying some content-related problems (missing links, inconsistent granularity) on the one hand, but allowed them to focus on areas where the developers' over-exposure may have dulled their sensitivity (navigation).

Two of the themes identified merit specific attention. Both groups identified the need to provide instructions for the users. Regardless of how "intuitive" a system is reputed to be, until the user knows what to do there will be a need for instructions. In this particular case the need was identified in relation to context-sensitive maps -- a brief phrase such as "click on the area of interest" may not be sufficient for the first-time user. This is where the power of hypertext could effectively be used to provide a multi-level help system that would give the novice user a great deal of explanatory information, but could be ignored by the expert user.

The second theme of particular interest has to do with chunking. If WWW systems are developed from scratch, with all documents being written specifically for the system, chunking should be relatively straightforward (though there is certainly room for disagreement as to ideal document length, the exact meaning of the word "topic", etc.) If, as in this case, existing documents are being retrofitted into a WWW system, the differences between print and on-line media will become painfully evident.

Since the evaluators were instructed to look for specific usability violations, as defined by the heuristics, it is not surprising that those were the violations they found. During the individual sessions, however, several evaluators found problems that they could not relate back to any of the heuristics they had been given. Though the experimenters had intended to allow evaluators to "invent" new heuristics during the evaluation sessions, this was not made clear during the initial briefings. Based on the results of the two groups, the experimenters have refined the list of heuristics, adding one more item, and rephrasing several of the existing items.

[Sidebar #7: Usability Principles (Heuristics): Revised]

Conclusions

Certainly, one obvious conclusion from our experiment is that developers should be given a project overview, a list of usability principles, and a style guide before they start work, rather than developing such documents after the fact.

Heuristic evaluation, as a process, was in fact easy to learn. One of the experimenters had attended a half-day workshop given by Nielsen at the CHI '94 conference; this, along with a review of Nielsen's written description [1], was sufficient to run the experiment. The evaluation was cheap, as no additional equipment was required. The evaluation was fast: each group completed its work in slightly over two days. The experimenters did find, however, that much time was consumed planning and scheduling the sessions, compiling and typing lists of problems, and analyzing the final results (this seems to be true for all types of evaluation).

One major weakness in our assessment of heuristic evaluation as a method is that we have no baseline (an empirical user-based test) against which to measure our results. Nielsen, Karat, and Desurvire [1] claim that heuristic evaluations detect between 40% and 60% of the usability problems an empirical user test would find, and also claim that the types of problems found are roughly comparable. We must trust that this is so.

Another weakness in this evaluation is that all testing was performed from the same machine (a 486 PC running Microsoft Windows) using the same browser (NCSA Mosaic Alpha 7), over the same local area network. Though this meant that the evaluators had a consistent platform, and thus their observations can legitimately be compared to each other, it does not reflect the diversity of platforms the ultimate user base will be using.

The most startling result of this heuristic evaluation was how similarly the UI experts and developers performed. This flies in the face of generally accepted HCI experience, and deviates from the comparative analyses performed by Nielsen, Karat, and Desurvire [1].

The experimenters have three possible explanations for the similarities:

1) The prototype was unpolished enough that even a relatively untrained group of developers had no difficulty finding large numbers of violations.

2) The two groups were not actually as different as we first thought. Both groups were composed of internal system analysts and economists, and both groups had subject matter expertise.

3) Since a developer was present during the UI experts evaluation sessions, her answers and comments may have introduced some element of bias into the process.

The experimenters do not claim that developers are as good as UI experts, nor that internal experts are as good as professional usability specialists. What we believe our experiment did show, however, is that internal resources, including developers, can be effective in identifying usability problems.

The method identified specific and detailed violations of general usability principles. It did not, however, identify larger issues of structure or organization. The experimenters believe this is inherent in the method, and suggest using other methods to get at such broader issues (see Nielsen and Sano [3]).

The UI experts tended to exceed the parameters of the test. They had difficulty distinguishing between the browser and the pages being viewed, so many comments related to things that were beyond the system developers' control. In addition, the UI experts displayed a tendency to criticize the content of the pages and the requirements of the system itself. They seemed to blur the line between UI evaluator and potential end-user of the system. Though insightful comments and helpful suggestions resulted from this, it also distracted from the examination of the system before them.

The evaluators' responses to the evaluation are instructive. The developers were quite enthusiastic about the process. Their ability to stand back and objectively evaluate their own pages, and those of their peers, without becoming defensive, was impressive. Three factors apparently play into their expressed desire to see such evaluations continue: given time and experiential constraints, most developers knew of areas in which the pages they developed were inadequate. The evaluation gave them an opportunity to document their objections in a safe environment. The developers also appreciated the opportunity to step back from their narrow implementation tasks and analyze the project as a whole. This made them feel more involved in the big picture, and also feel that their concerns and judgments were being treated seriously and with respect. Finally, it was a nice change from cutting code, and was enjoyable simply as something different to do.

The UI experts were more restrained. The method itself seemed to strike them as basically old approaches wrapped in new language. They were adamant that such evaluations not replace end-user testing.

During some parts of the UI experts' evaluations a developer was present (during the individual sessions the developer stood by to help with problems and answer questions, during the debriefing session a different developer took notes). At times, various criticisms clearly made these developers uncomfortable. This made some sessions more difficult to facilitate than would have been the case if developers had not been present.

In the long term, perhaps the most lasting result of the heuristic evaluation goes beyond the specific system tested, and relates to internal organizational development. For the last six or seven years, the Bureau of Labor Statistics has been going through a gradual, but noticeable, educational process concerning the importance of HCI. Analysts have been expanding their understanding of software development, beginning with a one-hour lecture in 1987, followed by a longer class, the establishment of the User Interface Working Group, the development and dissemination of HCI guidelines within BLS, regularly scheduled UI classes becoming part of the expected curriculum for software developers, and attendance at conferences and seminars. This heuristic evaluation was another step in this educational process. The experimenters and evaluators learned to use the method and to incorporate the results into subsequent development; the rest of the organization has been informed about the process through formal presentations as well as hallway

conversations. Explicit usability testing has entered the organizational consciousness in an unintimidating way, and future projects are now likely to include some form of this activity as part of the accepted life cycle of software systems.

Overall, the experimenters consider the heuristic evaluation process to have been a success. A large number of usability problems were identified with a reasonable expenditure of effort. We shall include the method as one tool in a collection that includes other inspection methods and end-user testing.

Addendum

The production BLS Web site, based on the heuristic evaluation described above along with other usability tests, was released on Labor Day, 1995. It can be reached at http://stats.bls.gov

Notes:

[1] *Usability Inspection Methods*. Jakob Nielsen and Robert Mack, eds. 1994 John Wiley and Sons, Inc.

[2] *Hypertext Hands-On! An introduction to a new way of organizing and accessing information*. Ben Shneiderman and Greg Kearsley. 1989 Addison-Wesley

[3] *SunWeb: User Interface Design for Sun Microsystem's Internal Web*. Jakob Nielsen and Darrell Sano. http://www.sun.com/technology-research/sun.design/sunweb.html

Sidebar #1

Project Overview

I. Objectives

To construct a public access system directly available to any user of the Internet with appropriate client software. It shall be optimized for ease of learning and ease of use by both first-time and experienced users, and shall encourage exploratory browsing. The system will:

- Explain the purpose and programs of the Bureau of Labor Statistics
- Provide BLS survey data in varying levels of detail
- Provide full explanatory information on BLS survey data

II. Intended Audiences

Due to its nature as a public access system, the intended audience is highly diverse, with a broad range of subject matter knowledge and computer skills. Specifically, users will vary in:

- Economic and statistical training
- Familiarity with the Bureau of Labor Statistics and its internal organization
- Expertise in using WWW browsers such as Mosaic or Netscape
- Hardware, software, and network platforms

III. Expected Usage Patterns

Corresponding to the diversity of the user base is the diversity of user needs. Some sample usage scenarios might be:

- An academic or government researcher looking for multiple data series spanning multiple surveys, but without any specific series in mind at the beginning of the search. This researcher also needs detailed background information on survey methodology and is interested in other relevant analytical papers or studies.
- An industry analyst looking for multiple data series from a single survey, knowing exactly which series she wants. This analyst retrieves the desired series every month and wants immediate and unencumbered access to her standard query.
- A journalist looking for a summary of a given survey's latest release along with an analytical overview.

- A high school student writing a social studies paper on the US economy, with no idea of what is available or what it means.

Sidebar #2

Usability Principles (Heuristics)

1. **Speak the users' language**. Use words, phrases, and concepts familiar to the user. Present information in a natural and logical order.

2. **Consistency**. Indicate similar concepts through identical terminology and graphics. Adhere to uniform conventions for layout, formatting, typefaces, labeling, etc.

3. **Minimize the users' memory load.** Take advantage of recognition rather than recall. Do not force users to remember key information across documents.

4. **Flexibility and efficiency of use**. Accommodate a range of user sophistication and diverse user goals. For example: guide novice users through a series of progressive steps leading to the desired goal, but allow expert users to directly reach their destination.

5. **Aesthetic and minimalist design**. Create visually pleasing displays. Eliminate information which is irrelevant or distracting.

6. **Chunking**. Write material so that documents are short and contain exactly one topic. Ideally, a document should fit on a single display page. Do not force the user to access multiple documents to complete a single thought.

7. **Progressive levels of detail**. Organize information hierarchically, with more general information appearing before more specific detail. Allow the user to delve as deeply as needed, but to stop whenever sufficient information has been received.

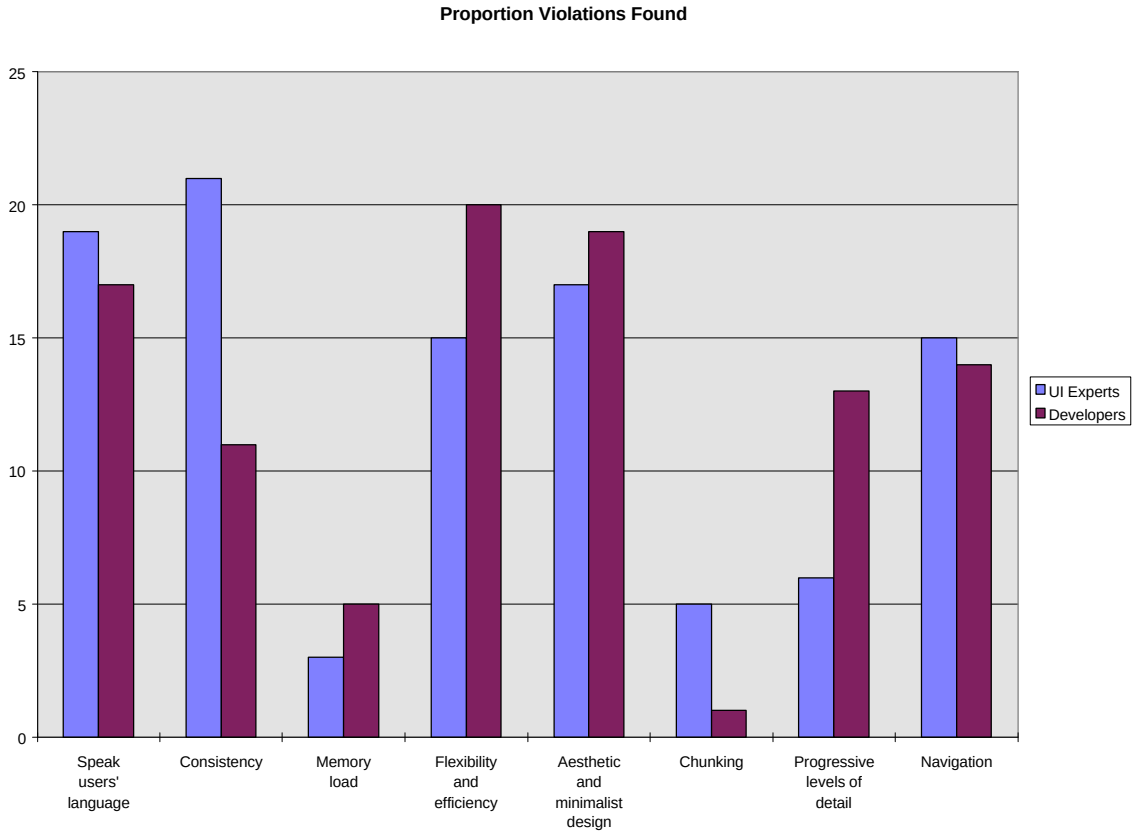
8. **Navigational feedback**. Allow the user to determine her/his current position in the document structure. Make it easy to return to an initial state.

Sidebar #3

Severity Rating Scale

0: I don't agree that this is a usability problem
1: Cosmetic problem only -- need not be fixed unless extra time is available on project
2: Minor usability problem -- fixing this should be given low priority
3: Major usability problem -- important to fix, so should be given high priority
4: Usability catastrophe -- imperative to fix this before product can be released

Sidebar #4


Proportion Violations Found
25
20
15
10
5
0
Speak users' language
Consistency
Memory load
Flexibility and efficiency
Aesthetic and minimalist design
Chunking
Progressive levels of detail
Navigation
UI Experts
Developers

Sidebar #5

Severity Ratings by Group

		UI Experts	Developers
Speak users' language	Avg. Severity	3.17	2.91
	Std Deviation	1.16	.68
	% Total Violations	19	17
Consistency	Avg. Severity	3.08	2.89
	Std Deviation	.99	.79
	% Total Violations	21	11
Memory load	Avg. Severity	2.92	3.00
	Std Deviation	.90	.74
	% Total Violations	3	5
Flexibility and efficiency	Avg. Severity	3.18	3.01
	Std Deviation	.77	.92
	% Total Violations	15	20
Aesthetic and minimalist design	Avg. Severity	2.25	1.92
	Std Deviation	1.52	1.12
	% Total Violations	17	19
Chunking	Avg. Severity	3.40	3.50
	Std Deviation	.68	.58
	% Total Violations	5	1
Progressive levels of detail	Avg. Severity	3.71	3.08
	Std Deviation	.55	.56
	% Total Violations	6	13
Navigation	Avg. Severity	3.32	3.22
	Std Deviation	.84	.67
	% Total Violations	15	14

Sidebar #6

Thematic Problems Identified

Heuristic	UI Experts	Developers
1: Speak users' language	Use of jargon Uninformative ordering of lists	Not enough information Misleading titles
2: Consistency	Terminology Link term *vs.* Page header Formatting (typeface, header, graphics, layout) Button labels ('go', 'run', etc.)	Formatting (typeface, header, graphics, layout)
3: Memory load	No theme	No theme
4: Flexibility and efficiency	Need instructions Difficulties finding desired material Insufficient short cuts	Need instructions Optimize 'applications'
5: Aesthetic and minimalist design	Visual appeal Redundant objects on screen Missing information	Visual appeal Position elements for visibility
6: Chunking	Separate topics merged Same topic split	Separate topics merged
7: Progressive levels of detail	No theme	Insufficient detail Inconsistent granularity
8: Navigation	Insufficient navigation aids (titles, headers, etc.) Inaccurate or unclear links	Missing links

Sidebar #7

Usability Principles (Heuristics)
Revised after Evaluation

1. **Speak the users' language**. Use words, phrases, and concepts familiar to the user. Present information in a natural and logical order.

2. **Be Consistent**. Indicate similar concepts through identical terminology and graphics. Adhere to uniform conventions for layout, formatting, typefaces, labeling, etc.

3. **Minimize the users' memory load.** Take advantage of recognition rather than recall. Do not force users to remember key information across documents.

4. **Build flexible and efficient systems**. Accommodate a range of user sophistication and diverse user goals. Provide instructions where useful. Lay out screens so that frequently accessed information is easily found.

5. **Design aesthetic and minimalist systems**. Create visually pleasing displays. Eliminate information which is irrelevant or distracting.

6. **Use chunking**. Write material so that documents are short and contain exactly one topic. Do not force the user to access multiple documents to complete a single thought.

7. **Provide progressive levels of detail**. Organize information hierarchically, with more general information appearing before more specific detail. Encourage the user to delve as deeply as needed, but to stop whenever sufficient information has been received.

8. **Give navigational feedback**. Facilitate jumping between related topics. Allow the user to determine her/his current position in the document structure. Make it easy to return to an initial state.

9. **Don't lie to the user.** Eliminate erroneous or misleading links. Do not refer to missing information.