

Beating the Rap on UI Standards

— published in *The X Journal*, March/April, 1996



Eric Schaffer, Ph.D., CUA, CPE, is Founder and CEO of Human Factors International, Inc. (HFI). He teaches, consults, and speaks on corporate and governmental Web and GUI interface design issues.



John Sorflaten, Ph.D., CUA, CPE, teaches and consults as a Project Director at HFI. With Eric, he initiated a usability curriculum at a local university in his home town of Fairfield, IA.

©1996, Human Factors International, Inc.

Remember when your folks asked (told) you to clean out the garage? Or the closet—if you lived in a Manhattan apartment. Even though you ate a candy bar for energy to finish the job, did you get a “No-win” proposition? Your parent (guardian, brother, sister, etc.) always came back with “But this should have gone there!”

THE RAP YOU ALMOST CAN'T WIN The unspoken appellation “you dolt” always remained in the air. What a rap. It was no win as long as you did it by yourself. However, when the enlightened parent (guardian etc.) did it with you, they shared responsibility, regardless of the time and effort. Bottom line: no finger pointing, ergo, no bum rap.

When your manager asked (told) you to write a user interface (UI) design standard, was it a no-win proposition? Apparently many developers feel that way. In our GUI design seminar, we invite participants to join in some “group therapy” venting. Here’s a recent list of reasons why standards don’t work:

- Written by people other than me!
- Often too many standards to remember
- Fuzzy: guidelines vs. standards
- Creates biases
- Problems with propagation among developers
- Too general for certain tasks
- Version problems
- When every component of the interface is “standardized” this effort can be overwhelming
- No creativity
- Tedious
- Hard to enforce
- Hard to keep up to date
- Costly to create
- Too difficult to change
- Too specific to certain platforms

What to do? Read on. Let’s hear from two project managers that have survived the rap. We met these managers while helping them solve their standards problems. In dealing with numerous large firms, we’ve handled every issue including cross-platform and international require-

ments. As you'll see, beating the rap requires a solid process as well as solid ergonomics. Get management and users involved.

CASE STUDY #1: ROYAL BANK OF CANADA

Royal Bank fielded 13 members on their standards committee. Additional “reviewers” allowed working with a larger audience. Royal’s Jamie Ingram provided project coordination. HFI’s Dr. Eric Schaffer headed the project.

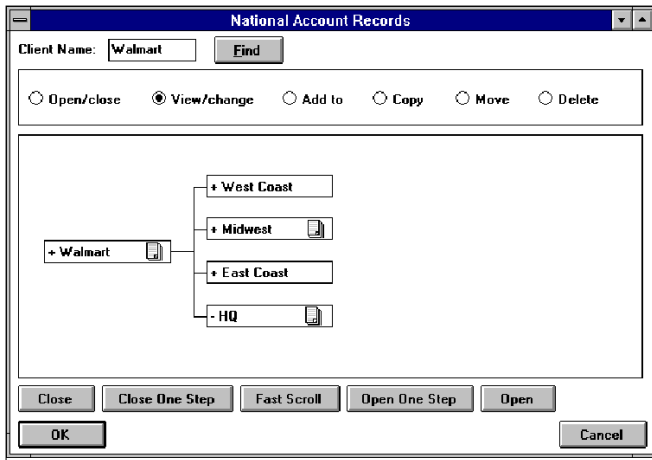


Figure 1. Sample “Browser” screen type that facilitates hierarchical relationships. A very powerful screen.

Q: How did you insure that your standard would be practical?

Jamie: We focused the standard on our business users. Our emphasis was on reducing the amount of training and relearning necessary to use our applications. The first step in the process was to collect data about our current business applications. For four days we reviewed our business applications with developers and users alike. We then created a

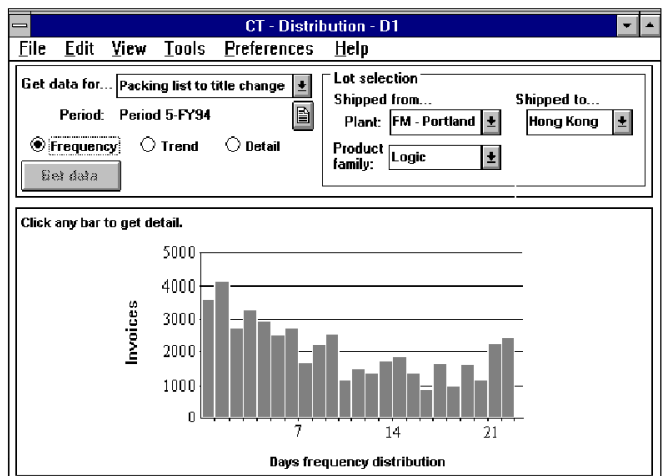


Figure 2. Sample “Graphic Drilldown” screen type. Remind developers to use graphic screens, especially when they come from a text-based environment!

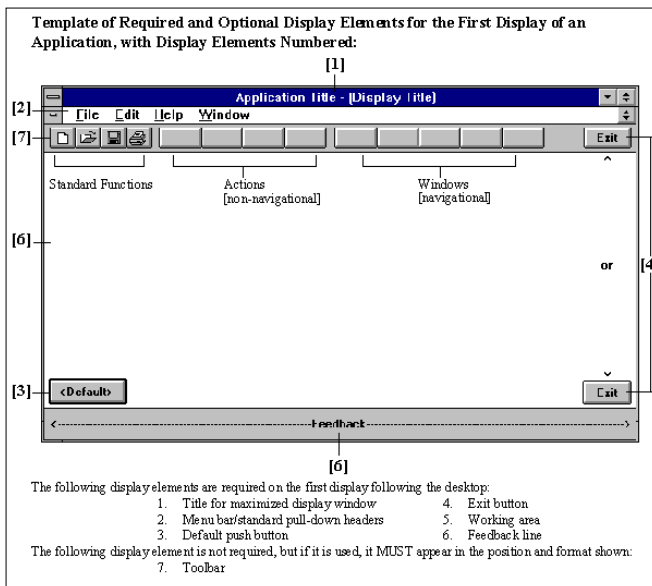


Figure 3. Sample “elements that appear on every screen.” Callouts provide links to the textual checklist.

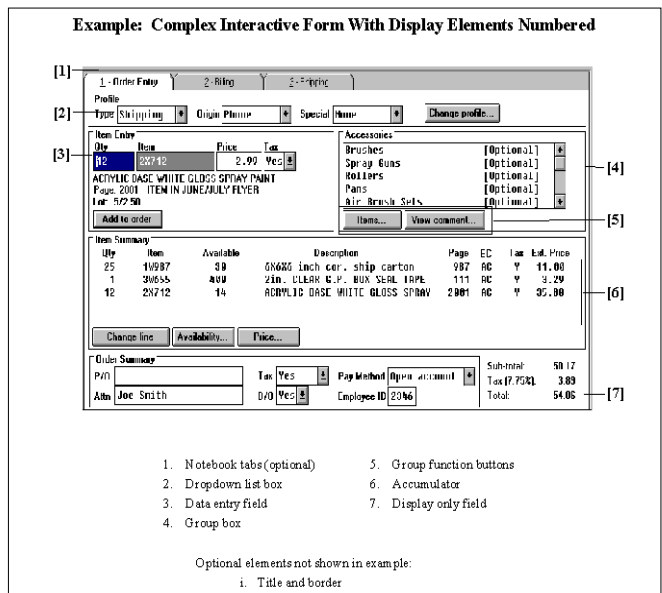


Figure 4. Sample “Form” screen type. Again, callouts link to a brief textual checklist.

set of standard screen types. (See Figures 1-4.) We used actual Royal Bank applications as examples to demonstrate each screen type.

Q: How did you handle problems of varying interests and opinion groups?

Jamie: We built in a wide consultation and review process. “We” is the operative word here. “We” refers to a long list of Royal Bank staff from a variety of groups. It also includes input from an external usability firm (HFI). Even before the first meeting, the committee members participated in a three-day course on graphical user interface from HFI. Then, over a period of four months, we met for several grueling one- and two-day meetings. The committee structures and defined the early drafts of the standard.

Q: How did you keep the document from intimidating and boring developers?

Jamie: Separate sections deal with specific tasks. Developers do not have to read right through every page from cover-to-cover to make effective use of it. Developers can quickly flip to a screen type and see a Royal Bank example. Behind each example are instructions on when to use that type of screen, a checklist of relevant standards for display and operation.

Q: What follow-up do you have for the standard?

Jamie: A new group has been formed to help applications develop their user interface. They will be building standard reusable code. This will make following the standard easier. They are also building working examples of each of the standard screen type examples. A short course introducing the Royal Bank GUI standards will be available shortly. This course will focus on how and when to use the standards. [Developers] can also take the same course that each of the committee members

took. This will help [them] understand some of the more specific human factoring principles and research that went into the standard.

Q: What benefits do you see coming from your standards effort?

Jamie: Less time spent evaluating design alternatives. More time for business function specific design work. Easier porting of system modules. Easier system maintenance. Less time providing phone support to users.

CASE STUDY #2: LARGE INTERNATIONAL CROSS-PLATFORM FIRM

This international organization fielded 15 members on their standards committee. It encompassed both Motif and Windows. The IS manager preferred anonymity. We’ll call her Adamantine. The following is based on several interviews. HFI’s Dr. Eric Schaffer and Dr. Diana Nelson both headed the very large project.

Q: What were some special problems with this project?

Adamantine: One of the big issues for us is the lack of widgets in Motif. We used lots of Windows widgets, and our developers were concerned about how to develop equivalent ones and make them work in the Motif/UNIX world. We reminded them they only had to do that once, but they worried about it. Some managers resisted the expense of creating the Motif equivalent widgets. But they failed to take into account the increased productivity of users with the standard objects.

Q: What’s an example of a custom widget?

Adamantine: If you have a scrolled list with more than one column, users need help scanning left to right across the columns. Research shows that users make less mistakes with a blank line every five or six lines. It serves to guide the eye like a ruler. Our Motif scrolled lists didn’t support that feature.

Q: Did your people agree to creating the custom Motif widgets?

Adamantine: Yes. However, as a tactical compromise, we allow current projects to omit the precise tools for an interim period. However, the application must emulate the “behaviors” given by the standard. With our widely distributed development groups, including contractors, we may have a problem enforcing the standards until we get our custom Motif widgets. But I’m taking the five year point of view now. First get a standard, which we have. Then a toolkit, which is coming. Also, we’ll give education to the developers. Then we’re set until the next big interface development. Maybe virtual reality?

Q: Any other big problems?

Adamantine: Long ago, I felt that the process was the most important ingredient. [See table “The Right Way to Beat the Rap”] We tried doing a standard before, using a different outside contractor who failed to follow the right process. That standard dwelt on principles you get from a books and research authorities. It bordered on being a “generic” standard you can get from the bookshelf. But we failed to give our users screen design options that were meaningful to them. We didn’t look at the business-specific tasks performed by our many users. Consequently, the developers couldn’t use the standard effectively. There was too much reading. Too little connection with actual tasks in our business. We need screen examples that promote performance accuracy, ease of use, and task predictability.

Q: How did developers respond to the new approach at first?

Adamantine: We’ve discovered there is a global lack of understanding for the need of a UI process in our organization. We need to constantly battle

against corporate culture and “ideology”. We have a lot of 3270 developers and users who have never seen a custom GUI application that really accommodates a “workflow”. Instead, they use the “windows paradigm” they have seen (word-processing, spreadsheet, and development tools). But these models lead to window thrashing. Now we know that customer service representatives don’t want to “bounce around” with that kind of design. We really need to teach people something they think they already know how to do! This is hard. Our people just didn’t know what could be done! And they’re not familiar with potential productivity gains available if you get the right screen types.

Q: Did you overcome developer resistance? Especially given a second go-round on standards?

Adamantine: Doing it a second time was a definite challenge. We faced emotional exhaustion in our team. However, three strategies made it work.

- First, the committee consisted of developers who were actively working on projects. The standard had to be usable to them. No abstractions allowed in the standard. They felt involved, and therefore were committed. Plus we had good consultative leadership with concrete alternatives and trade-offs for us to evaluate.
- Second, the user representatives became strong allies. They helped set policies and requirements. They reviewed the screen types we had obtained in prior interviews with other users. We educated our user reps in UI design so they could talk the same language as our developers. Previously, they hadn’t known their options, and thus failed to articulate what they didn’t like. They were pleasantly surprised at their options.
- Third, we had a large, important project that needed the ideas we generated during the standards process. It became a showcase project for

other projects to emulate. It was a big motivator because the showcase project demonstrated concrete productivity increases from the standard.

Q: Any advice for others doing a multi-platform standard?

Adamantine: Be selective. Don't try to standardize everything. For example, don't bother standardizing button sizes. You can't predict all situations or CRT sizes and resolutions. Don't standardize on a single color scheme. Instead provide several preplanned color palettes from which developers and users can select. We couldn't standardize on accelerator keys except for "Alt-X" for Exit, and Escape for Cancel. Previous applications and future situations varied too much across applications to allow such niceties. We were also limited in standardizing on F-keys except for Help and Exit. Obviously, other organizations would benefit if they could standardize at least certain F-key functions. In general, we had to create "function

tables" that allowed us to compare features across both Motif and Microsoft Windows platforms. We identified commonalities. Then we had to make hard decisions about non-overlapping functions. In some cases we used text to describe differences, or we excluded standards on contradictory features like different F-key assignments.

Q: What areas were ripe for standardization?

Adamantine: We found that the major benefits of standards applied across platforms easily. These included special over-all screen designs that handled our special task flows. In the screen types we selected, our developers got phraseology for instructions, examples of indentation, and important navigational ideas. They got a lot of "cosmetic" tips like the design of groups with headers and use of particular icons for warning, system failure, and informational dialogs. We standardized on the fact that the DEFAULT button should appear at the lower left, and this is not necessarily the OK button!

THE RIGHT WAY TO BEAT THE RAP

Here is an outline of the HFI GUI standardization process. You can do it, too. We've helped beat the rap on standards. It works, or we wouldn't have a company any more. Send us your experiences in standards. We're looking for anecdotes and screen pictures for a book on UI and Interactive Voice Response design standards. (We'll provide anonymity for the modest!)

Steps to Beat the Rap	Comments
1. Define the type(s) of standards you need. Keep them separate.	For example: Design; Methodology; Help; Error message; Interactive Voice Response; Imaging.
2. Get management commitment to follow through on the standardization process.	If necessary, get HFI's management briefing to "raise standards consciousness," or take a GUI design class. Don't trust anyone under 30. (Humor)
3. Get 7-12 volunteers for committee membership.	Get key opinion leaders. Volunteers work harder.
4. Gather data.	Cover user characteristics, taskflows, work environments, software usage, corporate strategies, etc. Conduct interviews at actual user worksites. People love attention. They'll like you, too.

- | | |
|---|---|
| 5. Define screen types that cover 85% of the windows that will be designed. | Locate important and prevalent tasks associated with your business. Customize a standard approach to handling those tasks. We define as many as eight primary screen types. You get efficiency and ease of use because a given type of task is handled with a standard configuration of objects. |
| 6. Draft an actual case for each screen. | Examples: Classical Hierarchical Menu, Browser (see Figure 1), Create-Review-Update-Delete (CRUD), Graphic Drill-down (see Figure 2), and Form (see figure 4).
Your standard should specify important elements of wording, layout, format, and operation of each screen. Here's your creative opportunity. |
| 7. Meet with the committee to refine the standard screen designs. | One or two days of grueling meetings, during which we iterate each prototype screen design with the group and discuss numerous design details. The committee votes on which screen examples to standardize. |
| 8. Draft the standards document around the examples. | Based on decisions made at the previous meeting, we might add new primary screen types, secondary screen types, popups, and dialog boxes. We'll further customize screen types to match the corporation's functional needs. We document rules associated with screen operation. (See Figures 3 and 4.) |
| 9. Add general chapters. | We document the rules for standard keys, color, error handling, icon usage, and widget selection. |
| 10. Meet again with the committee to review and approve the standards document. | This one or two day meeting includes feedback from external reviewers, as well. Gruel may or may not be served. |
| 11. Finalize the document. | We make the changes. We complete the Table of Contents, Index, and Glossary. We fix the document headers, section numbering and pagination. |
| 12. Implement the standard. | This phase may or may not include an additional iteration and meeting. |