Help!

Arbitration of a Help System

Garett Dworman, PhD | Tec-Ed, Inc. | garett@teced.com

DESIGN CAN BE AS MUCH AN ACT OF NEGOTIATION as it is of creation, and designers can find themselves cast in the role of negotiator or arbitrator among competing interests. Designers of user-assistance systems often face the most difficult negotiations, because user assistance tends to be perceived as a less important module. This secondary status means that proponents of user assistance negotiate from a weak position with respect to other modules' interests. This article presents a case study describing the design of the user assistance—and that design's negotiation—for an enterprise business's intelligence application.

One of my clients is a large company that is building an internal, enterprise business-intelligence portal that provides consistent and structured access to quantitative metrics from across the organization. The portal is built on top of Siebel Analytics (the Cayman release of 2005), which creates dashboards of query reports but is not a portal platform. While Siebel Analytics provides a powerful back-end engine for integrating data sources into a common business model, its limited UI prevents almost all inline help techniques and JavaScript customizations. Therefore, all user assistance must be provided off-page. Our job was to implement user assistance within the requirements and constraints of both the stakeholders and the platform.

The portal is designed with the following goals:
1. Provide a common user experience across all dashboards.
2. Provide a dashboard organization that enables users to easily navigate to known dashboards and discover unknown dashboards that can help them.

The portal requires two levels of user assistance:
1. Global Help for the portal itself
2. Local Help for each individual dashboard

In both levels, help includes instructive information to help users perform needed actions as well as supplemental information such as change-management processes, metric and domain ownership, known issues, and future plans.

DESIGN ITERATIONS. Figure 1 shows the page template for the portal's pages. Figures 2 through 5 show four design iterations of the simple user-assistance mechanism within that template:

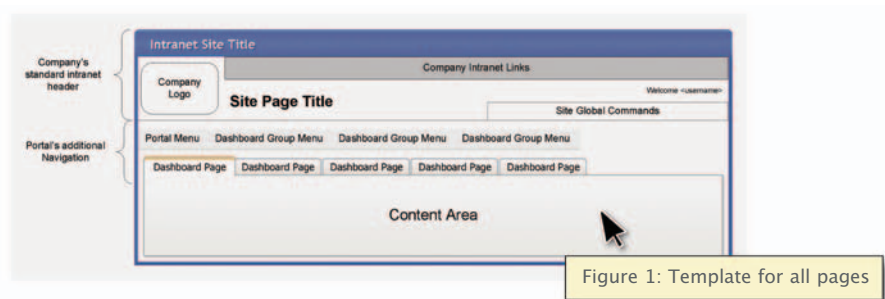Figure 1: Template for all pages

Figure 2 shows the Portal Design Team's initial design with equal emphasis on portal-level and dashboard-level user assistance. All user-assistance content is housed in a special

Figure 2: Iteration 1 design: Portal-
and dashboard-level user assistance
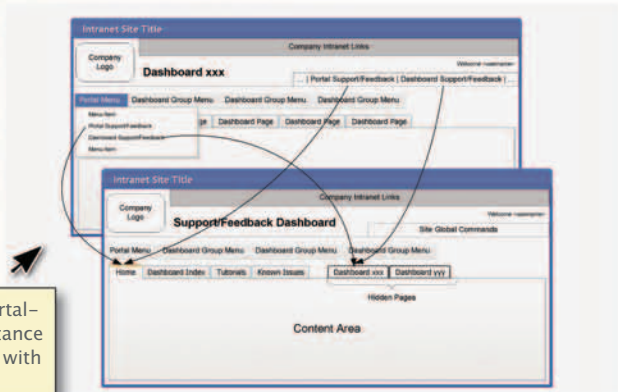integrated into one mechanism with
equal access to each.

Figure 3: Iteration 2 design:
Dashboard-level user assistance
provided within the dashboard,
portal-level user assistance provided
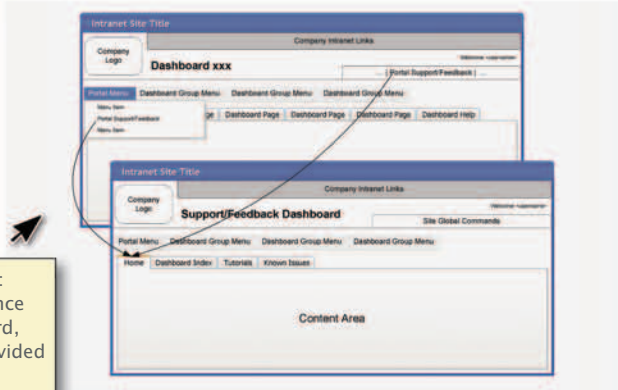in the external mechanism.

Figure 4: Iteration 3 design.
Dashboard-level help access now
provided by many external
mechanisms, and dashboard-level
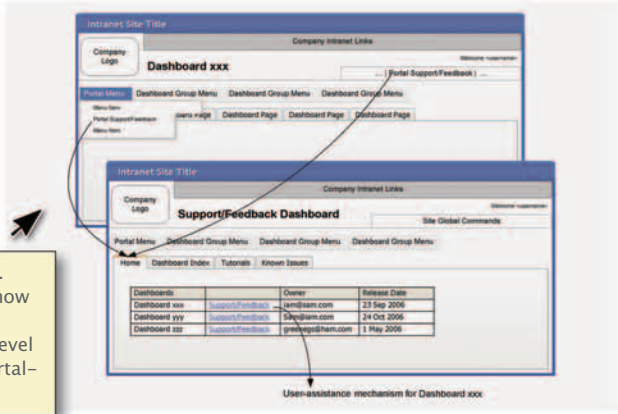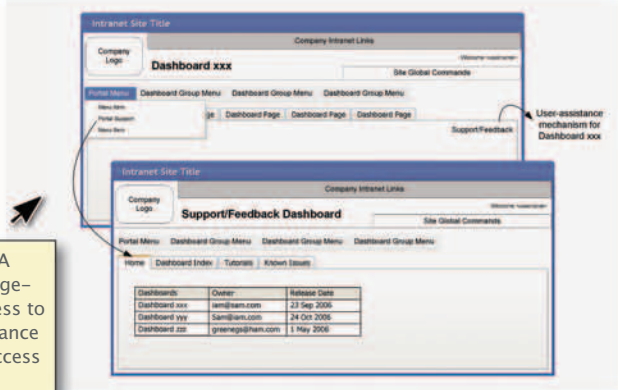access is submerged within portal-
level access.

Figure 5: Iteration 4 design. A
context-sensitive link in the page-
content area provides direct access to
the dashboard-level user-assistance
mechanisms, and portal-level access
is provided in the menu.

Support/Feedback Dashboard. Portal-level help is divided into dashboard pages accessed via Siebel Analytics tab navigation; dashboard-level help is provided on hidden pages, one per dashboard. The header and menu links go directly to the Support/Feedback Dashboard's Home page for portal-level user assistance, and the current dashboard's hidden page for dashboard-level user assistance.

The design in Figure 3 takes dashboard-level user assistance out of the global Support/Feedback Dashboard and puts it directly into the individual dashboards as the last tab in the dashboard. This design change accommodates a requirement from the company Internal Standards Body stating that the page header may have a single help link, that the link's label be exactly "Support/Feedback," and that the link be context-sensitive. Unfortunately, Siebel Analytics cannot provide context-sensitive links in the page header. Therefore, the one link is hard-coded to access portal-level help, and dashboard-level help remains in the dashboard.

In Figure 4, dashboard-level help is removed entirely from the portal. The manufacturing business unit insisted that all manufacturing dashboards leverage the hierarchical FAQ-tree, user-assistance mechanism used in the Manufacturing Business Unit's intranet. Since other business units could not use this mechanism, the portal needed to allow each business unit to use whatever mechanism they wanted for dashboard-level user assistance. Local dashboard-level user assistance is accessed via a list of dashboards in the Support/Feedback Dashboard's Home page.

Not surprisingly, users rebelled against the two-step process for getting to dashboard-level user assistance. Furthermore, the list of dashboards was not scalable. Figure 5 displays the latest design, which:

- has no Support/Feedback link in the portal header area—the Internal Standards Body allows there to be no Support/Feedback link
- has a Portal Support link in the portal menu that links to a portal-level help page. Portal menus are unique in the company's intranet, so the Internal Standards Body has no relevant standards, but they did require avoiding the reserved phrase "Support/Feedback" in the menu
- has a dashboard-specific Support/Feedback link in the content area of all pages of every dashboard that links to the dashboard's user-assistance mechanism

CHALLENGES AND RESOLUTIONS.

Raise help systems from a secondary priority. "An interface, as far as is possible, should be self-teaching...Help displays are simply part of the content. No special mechanisms or techniques are required to use them [5, p. 175]." The secondary status of user-assistance mechanisms makes this ideal difficult, if not impossible, to achieve. Organizations understand the ROI of well-designed and -executed user assistance, yet ignore their own advice and do not dedicate the resources necessary to augment the product as well as it could be [7].

One of the ways that this secondary status hurts the user-assistance module is that the stakeholders seldom want to "waste" the time for the needed arbitration process. The design iterations discussed here took almost a year, although each iteration took only a couple days' effort.

Moreover, the secondary status creates a weak negotiating position, often resulting in concessions to other system modules. The end product tends to be a tool with a poor balance between functionality and functionality support.

Make the arbitration process explicit. "Organizations adopt and ignore online help systems and features based on ... many incentives... [2, p. 287]." These often-conflicting incentives are the basis for the design negotiations. Plan for the negotiations as soon as you identify competing interests. Don't simply let them "occur" or you will waste many resources going back and forth between parties. Get the negotiating parties into a design

REFERENCES   1. Carroll, J. M. (ed), (1998) Minimalism beyond the Nurnberg funnel. Cambridge, MA: The MIT Press. 2. Covi, L.M. & Ackerman, M.S. (1995) *Such easy-to-use systems! how organizations shape the design and use of online help system* (pp 280 - 288). New York: ACM Press. 3. Grayling, T. (2002). A usability test of two browser-based embedded help systems. *Journal of the Society for Technical Communication, 49(2),* 193-209. 4. L. Kantner, S. Rosenbaum, and R. Shroyer, (September 2002). Structured heuristic evaluation of online documentation. From the Proceedings of the IPCC 2002, Portland, Oregon: IEEE. 5. Raskin, J. (2000). The humane interface, Boston: Addison-Wesley Professional. 6. Rintjema, L. & Warburton, K. (1998). Creating an HTML help system for web-based products. From the Proceedings of the Sixteenth Annual International Conference on Computer Documentation, Quebec City, Canada: ACM Press. 7. S. Rosenbaum, L. Kantner, & G. Dworman (July 2005). Helping users to use help: results from two international conference workshops. From the Proceedings of the IPCC 2005, Limerick, Ireland: IEEE. 8. W. Quesenbery (2001). Meeting user needs for useful online information. *Journal of the Society for Technical Communication,* 48(2), 182-188.

session, telling them it is "an opportunity to balance all parties' interests." It was arbitrated design sessions like these that achieved each compromise in iterations two through four.

Brainstorm designs with the stakeholders before mocking up. One result of the secondary status of user assistance is a lack of funds for field research on stakeholder needs and expectations for user assistance. This limitation means designers risk creating a user-assistance mechanism before obtaining a good understanding of the stakeholders' interests. The initial mockups are better informed, and additional interactions are shortened or eliminated, after at least a quick, two-hour brainstorming session with the stakeholders before creating mockups.

Make it clear that negotiation is not necessarily competitive. Negotiating a design, especially one for user assistance, does not need to be competitive. Of course, compromise is required, but usually a design that satisfies all parties can be found. Often, such a design is better than any that would have fit one party's ideal before the negotiation. When discussing the difficulties users had with the third iteration, the internal-standards body conceded that the page header did not need a Support/Feedback link, allowing the more intuitive, context-sensitive link in the content area of iteration four.

Make it easy for parties to give in. When two parties' interests are contradictory, make it easy and face-saving for one of those parties to give up on one of its interests. Explain how the resulting design is better for everyone, including them. It's almost always possible to find and emphasize an advantage to the sacrifice that makes it palatable. While the portal design team wanted a single, consistent, dashboard-level user-assistance mechanism, giving up this interest allowed for leveraging ready-made user-assistance mechanisms, saving resources that were needed to create new dashboards.

This diplomacy will be especially important in making sure the user assistance is properly socialized and that the sacrificing parties will be part of the effort to publicize it and encourage its use. Making the sacrifices easier is also important for ensuring that all parties will cooperate in future arbitrations—a design is never finished, simply temporarily frozen for the current release.

Use an iterative design process. Sometimes it is possible to evaluate all the parties' interests up front and then sit down and create a design that balances those interests. A large body of existing literature provides advice on how to build help mechanisms based on principles of design and usability [1, 6, 7, 8] or research data [3, 4]. However, too often the environment's web of competing interests is overly complex. The parties' interests are not always known up front because parties may assume that everyone shares their interest; they don't know how to express that interest; or they simply don't know their interests until they see an actual design. An iterative design process will allow the negotiations to bring out those interests and find the best compromise among them.

CONCLUSION.   Design of a user-assistance module is often a negotiation from a weak position because of the secondary status that businesses tend to give the user-assistance module. Therefore, if designers of user-assistance mechanisms cannot upgrade the priority of the user assistance from its secondary status, they need to expect an iterative process that allows for successive negotiation steps. There is no reason for negotiations to be competitive and ugly. Instead, the negotiations should be regarded as an opportunity to create the design that best meets the needs of all the interested stakeholders.

ABOUT THE AUTHOR *Garett Dworman, PhD, is a senior consultant at Tec-Ed, Inc., specializing in enterprise knowledge management. He has published extensively on how people and organizations disseminate, access, and consume information. His projects include analysis and design of business intelligence portals, business process and workflow tools, information integration applications, and network management applications. Garett earned his doctorate in decision sciences from The Wharton School, University of Pennsylvania, and undergraduate degrees in cognitive science from Tufts University.*