The Bull's-Eye: A Framework for Web Application User Interface Design Guidelines

Betsy Beier
Principal User Interface Designer
betsy.beier@oracle.com
+1 650 633 8828

Misha W. Vaughan
Principal Usability Engineer
mvaughan@acm.org
+1 650 607 0652

Usability and Interface Design Department
500 Oracle Parkway, MS 2op10
Oracle Corporation
Redwood Shores, CA 94065 USA

ABSTRACT

A multi-leveled framework for user interface design guidelines of Web applications is presented. User interface design guidelines tend to provide information that is either too general, so that it is difficult to apply to a specific case, or too specific, so that a wide range of products is not supported. The framework presented is unique in that it provides a bridge between the two extremes. It has been dubbed the 'Bull's-Eye' due to its five layers, represented as concentric circles. The center of the Bull's-Eye is the Component layer, followed by Page Templates, Page Flows, Interface Models and Patterns, and Overarching Features and Principles. To support this approach, requirements were gathered from user interface designers, product managers, UI developers, and product developers. Also, usability testing of the guidelines occurred on several levels, from broad guideline tests to more specific product tests. The guidelines and lessons learned are intended to serve as examples for others seeking to design families of Web applications or Web sites.

Keywords

user interface guidelines, standards, corporate style guides, enterprise style guides, Web applications

INTRODUCTION

The Challenges

We faced challenges common to many companies attempting to create user interface design guidelines for a family of Web applications. We were attempting to design for multiple, Web-based software products across a variety of user profiles – with only desktop application guidelines as our reference point. We knew technological limitations would also impact our guidelines as we attempted to make them accessible, cross-browser compatible, and localizable. This paper will discuss the problems we faced, often shared by other companies, and how we overcame them.

Our first challenge was the state of Oracle's existing UI guidelines. They were focused on Java, not HTML, and were at the widget level, and so did not provide use-cases, multiple options, higher level component combinations (i.e., templates or flows), nor contextual examples to illustrate usage. Our attempts to use other guidelines as exemplars left our guidelines too broad to implement specifically and consistently across products.

Second, the new guidelines would need to work with an evolving technology – Web applications. Web application guidelines could draw from desktop UI guidelines, however this approach would be limited in its usefulness. Web applications are delivered via a browser that is dominated by different metaphors (e.g., page-centric) than desktop applications (e.g., windows and menus). Also, this HTML-based UI would be less interactive than desktop applications – in order to support cross-browser compatibility, internationalization standards, and universal access standards.

Third, was the scope of our Web-based products. We needed to design for 100+ Web-based products, developed across multiple domains, and serving a variety of user profiles.

Fourth, we lacked any sort of centralization or support to design or implement a set of guidelines. Many development teams were already hard at work rapidly inventing their own product-specific look-and-feel.

To meet the challenges of developing HTML UI design guidelines, we would have to produce guidelines that were flexible enough to use across the variety of cases and issues, but specific enough to meet each team's requirements.

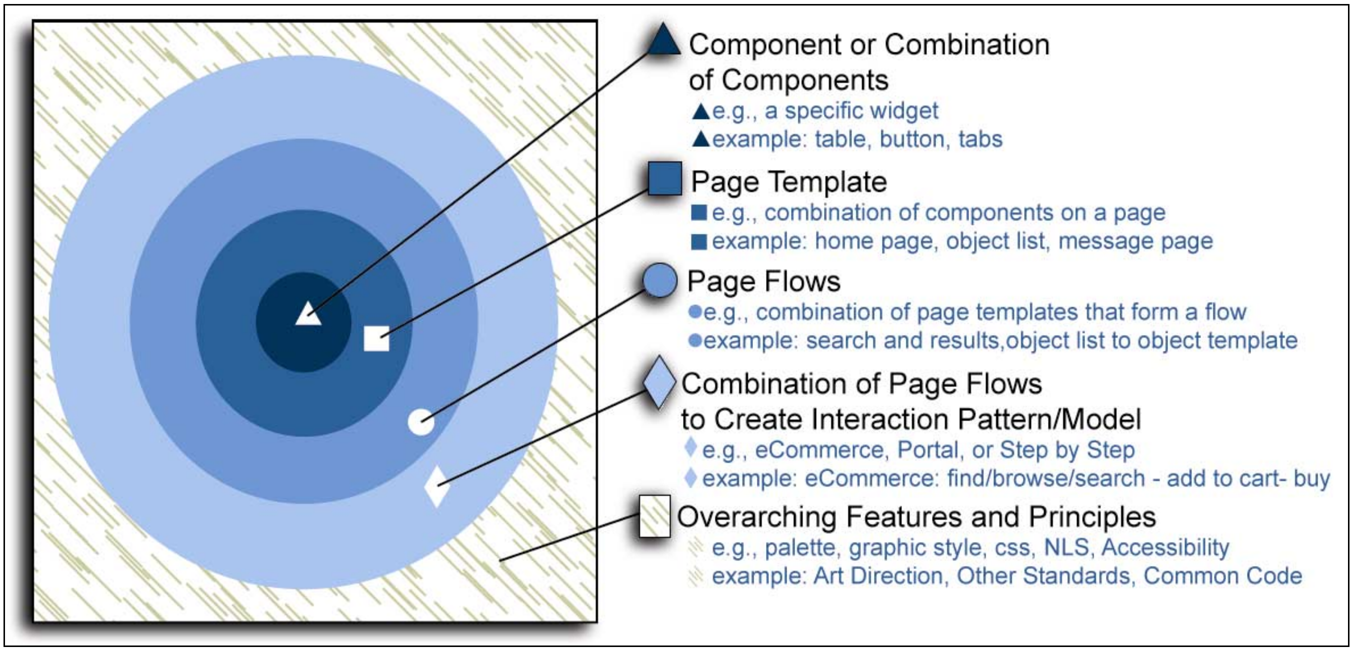
Our Solution – The Bull's-Eye[1]

The Bull's-Eye was designed to address each of the four challenges. The Bull's-Eye (Figure 1) represents the concentric circles of guidelines, each building on the next layer as one moves from the inside out. In the center of

CHI 2003, *April 5–10, 2003, Ft. Lauderdale, Florida, USA.*
Copyright 2003 ACM 1-58113-630-7/03/0004…$5.00.

[1] A 'bull's-eye' refers to a small, circular target in archery sporting events.

Figure 1. The Bull's-Eye: A Framework for Web Application UI Design Guidelines


▲ Component or Combination of Components
▲e.g., a specific widget
▲example: table, button, tabs
■ Page Template
■ e.g., combination of components on a page
■ example: home page, object list, message page
● Page Flows
●e.g., combination of page templates that form a flow
●example: search and results,object list to object template
♦ Combination of Page Flows to Create Interaction Pattern/Model
♦ e.g., eCommerce, Portal, or Step by Step
♦ example: eCommerce: find/browse/search - add to cart- buy
▨ Overarching Features and Principles
▨ e.g., palette, graphic style, css, NLS, Accessibility
▨ example: Art Direction, Other Standards, Common Code

the Bull's-Eye are Components, followed by Page Templates, and Page Flows. Interaction Models or Patterns sit on top of Page Flows. Finally, the Bull's-Eye is completed with Overarching Features and Principles.

Lack of an HTML UI Foundation

Before conceptualizing the Bull's-Eye and developing the standards, we evaluated our existing guidelines [10] as well as other software guidelines. An examination of approaches to developing style guides [2, 3, 7, 11], Graphical User Interface (GUI) guidelines [1, 4, 6, 16, 12, 13, 5], and Web style guides [9] demonstrated that existing resources were either too general or too specific. Existing approaches provided a set of user interface design principles (e.g., design guidelines and heuristics) that were too broad to apply in the specific case, or they provided a standard (e.g., at the UI component or widget level) that was too narrow to apply across the variety of cases faced by our teams. Unfortunately, we do not have room in this paper for a more detailed discussion of this analysis. This apparent gap in existing guidelines motivated our decision to create guidelines at multiple levels, from the UI component to overarching features. The notion of multiple levels of guidelines, from component level to flow level, became the basis for the Bull's-Eye metaphor.

Web Applications

As part of our investigation into Web design guidelines, we explored existing Web metaphors. We examined existing Web applications and Web sites, as well as elements from the desktop world that were still applicable in the Web browser environment. From this study, we developed a set of generalizations about new UI design guidelines for Web applications, such as: using a portrait page orientation (compared to a landscape page orientation), using browser-based hierarchical page organization, using free form layout (compared to GUI-style windows, panels, and toolbars), and using Web navigation structures such as tab navigation and side navigation (compared to menu structures and multiple dialog interfaces). We found that our generalizations were consistent with other, similar efforts by Web design consultants, e.g., Najjar's work at Viant [8], and consumer Web-application designers, e.g., Zhu's work at Microsoft [17]. We then applied our findings to each of the different levels of guidelines in the Bull's-Eye.

A second problem in designing for Web applications is that a browser-based UI would need to support cross-browser compatibility, internationalization, and accessibility. To accommodate this challenge, we chose to reduce the available interactivity in the UI – specifically, Javascript is used in only very limited contexts and Java applets are not used at all.

A Broad, Varied Product Base and Set of User Profiles

Our product suite ranges from server technologies, to development tools, to customer relationship management (CRM), to enterprise resource planning (ERP), to business intelligence. This broad set of products and their associated user profiles often spawned very different UI interpretations (Figure 2) of what were usually typical tasks across applications. For example, a task that crossed product boundaries, such as 'search,' looked and behaved differently from one application to another. The page flow that retrieved and displayed search terms and the search result set was inconsistently designed across applications. The display and navigation of primary objects (e.g., from purchase orders to database targets) also differed widely across applications; they shared no common look or interaction behaviors. Even straightforward tasks such as viewing reports varied greatly across applications. An examination of the product suites revealed UI inconsistencies across applications and divisions, and were most visible at the page and page-flow level. However, we knew it was possible to create a common look across systems [e.g., 5].

At the same time, there was a varied set of user profiles. For example, a self-service benefits application defined its users as 'any employee in a company who can use a mouse.' Every individual, from the janitor to the CEO, would need to be able to use this product to register for his or her benefits. At the other end of the continuum were database products, which defined their users as 'novice to expert database administrators' – who would have a much higher level of computing expertise.

To address the problem of needing a common look-and-feel across varied applications, we again turned to the notion of developing guidelines at multiple levels. The guidelines would not only encompass the individual UI components on a given page (e.g., the icons used on a search page), but also the page layout and a page flow (e.g., search page templates and a common search flow).

The Bull's-Eye, at this point, addressed the overall problem of creating a common user experience across a broad product group, but it did not address the issue of user variety within these groups. For instance a basic search page template and a search flow would satisfy many applications' needs (and potentially several user profiles), but would not satisfy all application's requirements, or varying user skill levels. It became clear that multiple options within each level of the guidelines would be needed to satisfy this variety.

Lack of Centralization and Support

In 1999, we received upper management support – a CEO-level mandate to move to a common look-and-feel. In order for this UI guideline and standardization effort to succeed, we needed three infrastructure practices in place:

- Education
- Coordination
- Communication

Figure 2. The Multiple Looks and Diverse Product Base of Oracle (circa 1997-1999)

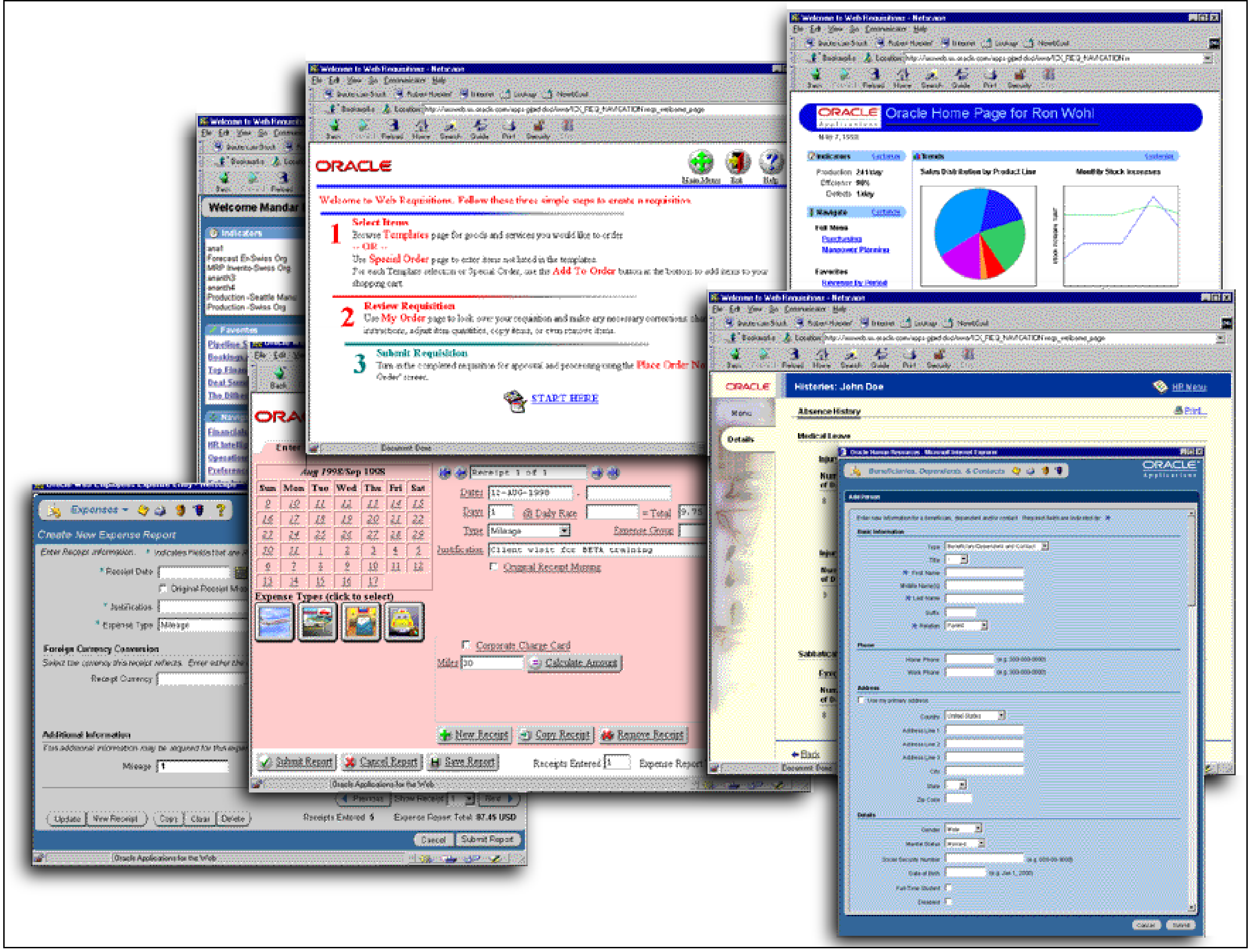
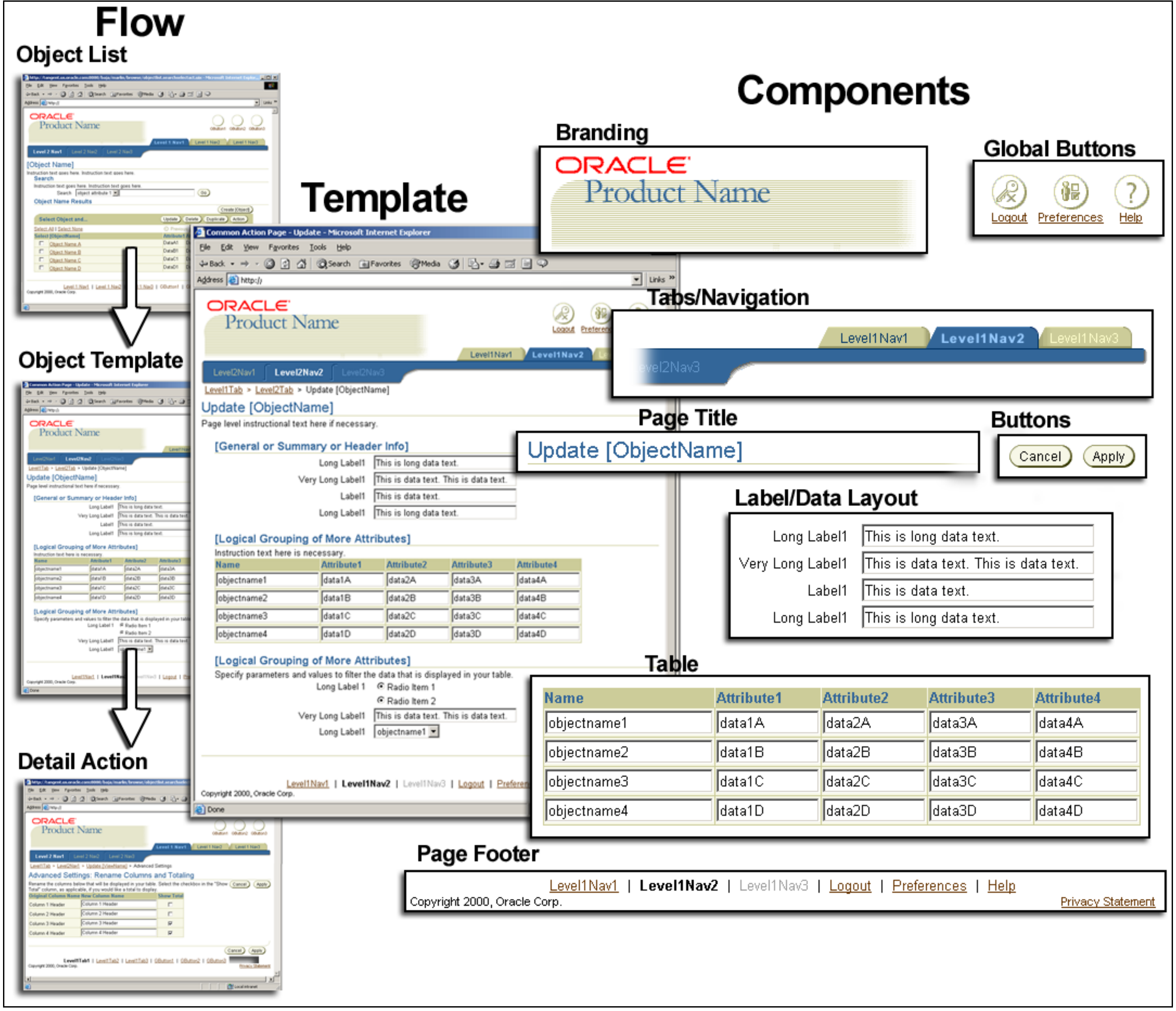

ORACLE
Main Menu
Exit
Help
Welcome to Web Requisitions. Follow these three simple steps to create a requisition.
1 Select Items
Browse Templates page for goods and services you would like to order
-- OR --
Use Special Order page to enter items not listed in the templates.
For each Template selection or Special Order, use the Add To Order button at the bottom to add items to your shopping cart.
2 Review Requisition
3 Submit Requisition
START HERE
Oracle Home Page for Ron Wohl
Indicators
Trends
Navigate
Sales Distribution by Product Line
Monthly Stock Increases
Welcome Mandar
Histories: John Doe
Absence History
Medical Leave
Details
Expenses
Create New Expense Report
Receipt Date
Justification
Expense Type
Mileage
Foreign Currency Conversion
Receipt Currency
Additional Information
Update
New Receipt
Copy
Clear
Delete
Cancel
Submit Report
Aug 1998/Sep 1998
Sun Mon Tue Wed Thu Fri Sat
Receipt 1 of 1
Dates 12-AUG-1998
Justification Client visit for BETA training
Expense Types (click to select)
Corporate Charge Card
Calculate Amount
New Receipt
Copy Receipt
Remove Receipt
Submit Report
Cancel Report
Save Report
Receipts Entered
Expense Report Total 87.45 USD
Oracle Applications for the Web
Oracle Human Resources
Beneficiaries, Dependents, & Contacts
Add Person
Basic Information
First Name
Last Name
Phone
Address
Details
Cancel
Submit

Figure 3. Sample Guidelines Components, Page Template, & Page Flow



Update [ObjectName]

Page level instructional text here if necessary.

[General or Summary or Header Info]

Long Label1	This is long data text.
Very Long Label1	This is data text. This is data text.
Label1	This is data text.
Long Label1	This is long data text.

[Logical Grouping of More Attributes]

Instruction text here is necessary.

Name	Attribute1	Attribute2	Attribute3	Attribute4
objectname1	data1A	data2A	data3A	data4A
objectname2	data1B	data2B	data3B	data4B
objectname3	data1C	data2C	data3C	data4C
objectname4	data1D	data2D	data3D	data4D

[Logical Grouping of More Attributes]

Specify parameters and values to filter the data that is displayed in your table.

Long Label 1 ● Radio Item 1 ○ Radio Item 2

Very Long Label1 This is data text. This is data text.

Long Label1 objectname1

Level1Nav1 | Level1Nav2 | Level1Nav3 | Logout | Preferences | Help

Copyright 2000, Oracle Corp. Privacy Statement

Education ranged from consulting with product teams to classroom lectures on the guidelines. In the early days of disseminating the information, the lead guidelines designer developed large-scale classroom lectures to introduce whole divisions to the guidelines. These lectures were often tailored to a particular domain, e.g., human resources. In an ongoing capacity, UI teams provided one-on-one consulting to product teams. This typically included translating a product's information architecture and task flows into a guideline's compliant interface design. Finally, a Web-based self-study course was created for the many developers we were unable to reach one-on-one.

Coordination took the form of: a) requirements gathering across products and user profiles, b) development of re-usable UI code, c) usability testing of the guidelines, and d) individual product reviews. Requirements gathering occurred weekly, and was an ongoing process designed to track needed changes, identify enhancements to existing guidelines, and push new needs into the next version of the guidelines. A re-usable UI code base was developed to ease adoption of the guidelines, and enhance consistency of implementation. A team was assigned within Oracle to develop re-usable UI code based on the guidelines. Close coordination and collaboration with this team has produced successful adoption of the guidelines code (how we define 'success' will be addressed later in the paper). To check the validity of guidelines, coordination with usability engineers provided the opportunity to conduct both product- and guidelines-specific usability tests. Finally, product UI reviews occurred at all levels, ranging from informal reviews with a UI designer to formal reviews with our department vice president.

Communication occurred both with the Usability and Interface Design department, and with the product teams they supported. Regular guidelines updates were disseminated via meetings and email notices to UI designers, usability engineers, product managers, developers, directors, and vice presidents. With this infrastructure in place, we were able to roll out the first version of the guidelines.

CONTENT & STRUCTURE OF THE GUIDELINES

We have laid out the challenges, as well as the framework and its multi-tiered approach, which is similar to other efforts to document UI patterns in web application flows and structures [e.g., 14, 15] A more detailed discussion of the structure and composition of the guidelines follows.

Components

Component guidelines, the first level of guidelines, outline specific UI widgets that have multiple interaction possibilities and options, depending on the content or functionality required by an application. Components can be simple UI elements, i.e., buttons, standard Web widgets, instruction text. They can also be more complex UI elements, i.e., tab navigation structures, table configurations and behaviors, and tree components, similar to the notion of *idioms* [13]). Figure 3 provides examples of several components covered in the guidelines, including Branding, Action/Navigation Buttons, Global buttons, Tabs and Navigation, and Tables. There are currently over 37 component guidelines, ranging from simple to complex. Examples of additional Components include Advertising, Locator Elements (e.g., Train, Breadcrumbs), Headers/Subheaders, Links, Content Containers, Hide/Show widgets, and a Page Footer.

Page Templates

At the next level are Page Templates. Page templates are comprised a combination of Components on a page. For example, an Attachments Page shows the UI components for situations when documents or notes need to be added to an object in an application. Within each Page Template guideline there may be multiple Component choices and layout options, each slightly different but all satisfying the same overall goal of consistency. The multiple options help satisfy the requirements for the broad range of users and applications supported. Figure 3 shows an example of an Update Page Template. This template indicates the placement of general components (e.g., Page Headers, Breadcrumbs) as well as specific components (e.g., attribute tables).

There are currently 30+ Page Templates in the guidelines. Examples of additional Page Templates include, Home Page Template, Overview/Summary Page Template, Search Page Template, Object List Templates, Object Templates, and a Step by Step/Wizard Page Template.

Page Flows

Page Flows are the third level in the Bull's-Eye, and consist of combinations of Page Templates (i.e., Add an Attachment Flow). This level of guideline outlines a combination of Page Templates with contents that form a common task flow. There are currently 20+ page flows in the guidelines. One example of a Page Flow is the Create/Add/Update/Delete an object Page Flow (Figure 3). This Page Flow handles the scenario of a user viewing a summary list of objects, then selecting an action and drilling down to the appropriate update, duplicate, or delete page. These are similar to notion of UI patterns

Other types of Page Flows include: Search and View results, Browse and Find an Object, Update Preferences, Manage Attachments, Export/Import, Customization of Tables, Intra-Application Navigation, and Inter-Application Navigation.

Interaction Models and Patterns

After Page Flows are Interaction Models and Patterns. These are groups of Page Flows that support a particular genre of application, such as ecommerce, portal, or administration applications. Each of these UI Models is a base consisting of a combination of common Page Flows and Page Templates, along with customizable aspects. For instance many ecommerce applications share a common set of tasks that can be supported by a base set of Page Templates and Page Flows. For example, the base UI model for an ecommerce application includes a Home Page Template, a Browse and/or Search Page Flow, an Item Detail Page Template, a Shopping Cart Page Template, a Purchasing Page Flow, and a Confirmation Page Template.

These models emerge as we work with teams to apply the guidelines and find a consistent set of commonalties. By creating these UI Models, we ease each team's initiation of their design process; we are able to indicate to a team the set of Page Templates and Page Flows that are most likely to relate to their application.

Overarching Features and Principles

Surrounding the entire Bull's-Eye are Overarching Features and Principles. These guidelines provide heuristics and standards that are used throughout the concentric circles to maintain a consistent user experience from component to UI Model. Examples of these Features and Principles include Object versus Action Orientation, Art Direction Standards, Language in the UI, and Accessibility Standards.

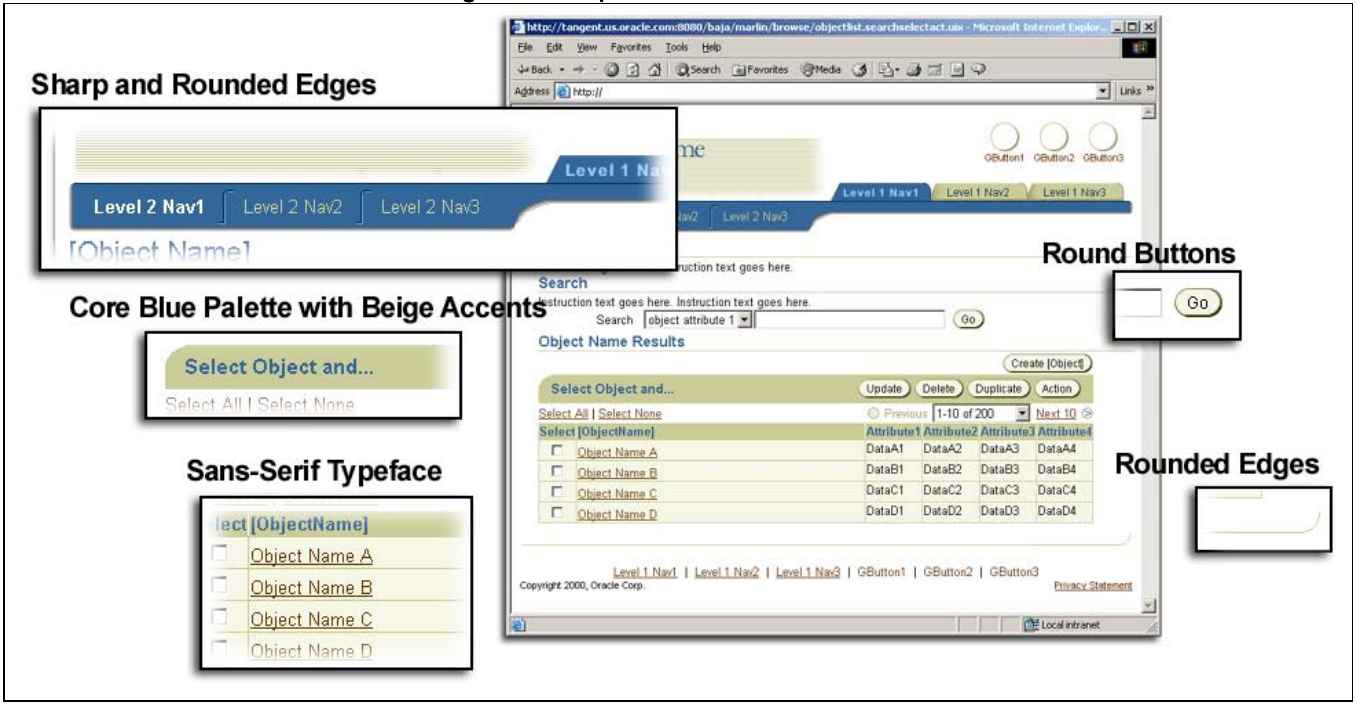
Object Versus Action Orientation

The overall structure of an application implementing the Bull's-Eye is an object-centric model. Content in applications is grouped around objects (or groups of objects) that form large functional areas of the application. Actions are surfaced from within the page contents, and/or the context of the object(s).

Art Direction Standards

The Art Direction standards (Figure 4) are composed of a Language Guideline, a Color Palette Guideline, Art Direction Guidelines, Ancillary Graphic Style Guidelines, and Cascading Style Sheet Text Standards. The highlights of the standards include:

Figure 4. Sample Art Direction Standards


Sharp and Rounded Edges
Level 1 Na
Level 2 Nav1
Level 2 Nav2
Level 2 Nav3
[Object Name]
Core Blue Palette with Beige Accents
Select Object and...
Select All | Select None
Sans-Serif Typeface
lect [ObjectName]
Object Name A
Object Name B
Object Name C
Object Name D
Round Buttons
Go
Rounded Edges
http://tangent.us.oracle.com:8080/baja/marlin/browse/objectlist.searchselectact.uix - Microsoft Internet Explor...
File Edit View Favorites Tools Help
Address http://
GButton1 GButton2 GButton3
Level 1 Nav1 Level 1 Nav2 Level 1 Nav3
Search
Instruction text goes here. Instruction text goes here.
Search object attribute 1
Go
Object Name Results
Create [Object]
Select Object and...
Update Delete Duplicate Action
Select All | Select None
Previous 1-10 of 200 Next 10
Select [ObjectName] Attribute1 Attribute2 Attribute3 Attribute4
Object Name A DataA1 DataA2 DataA3 DataA4
Object Name B DataB1 DataB2 DataB3 DataB4
Object Name C DataC1 DataC2 DataC3 DataC4
Object Name D DataD1 DataD2 DataD3 DataD4
Level 1 Nav1 | Level 1 Nav2 | Level 1 Nav3 | GButton1 | GButton2 | GButton3
Copyright 2000, Oracle Corp.
Privacy Statement
Local intranet

- A predominantly blue and white palette with beige accents
- Complementary round and square shapes throughout
- Visual hierarchies of information displayed through size and navigation depth
- Sans-serif typeface
- Minimizing rendered icons
- Simplified, stylistic graphics in content
- Common terminology and grammar rules

Support of Other Standards

The guidelines also had to adhere to other standards within the company. For instance, applications are translated into 28 languages. The guidelines must follow common translation and natural language support heuristics. Other standards that needed to be supported were the Federal Accessibility guidelines (i.e. Section 508 compliance), corporate Abbreviation guidelines, and corporate Keyboard Shortcut standards.

Common Code Standards

As mentioned earlier, in order for the Bull's-Eye to achieve its goal of creating a common look-and-feel across Oracle applications, it needed a unifying integration point with all applications — a common code base supporting the UI standards. Common code that supports each individual UI guideline greatly reduces variance and divergence from the standards. This code base was driven in part by the development schedule of the guidelines, but also by demand for the common code from all of the product teams.

The need for a common code base also drove, in part, the need to develop the guidelines from the inside out, i.e., starting with Components. It would be difficult to start coding common Page Templates or Page Flows, without having common Components already in place.

Structure of a Guideline

Each of the guideline levels mentioned above, Components, Page Templates, Page Flows, UI Models and Patterns, and Overarching Features and Principles, are communicated in a consistent format (Figure 5). This format eases use and maintenance of the guidelines.

Each guideline is documented in HTML, and posted as part of a complete guidelines Web site (publicly available at: http://otn.oracle.com/tech/blaf/). Each guideline is composed of several sections, including:

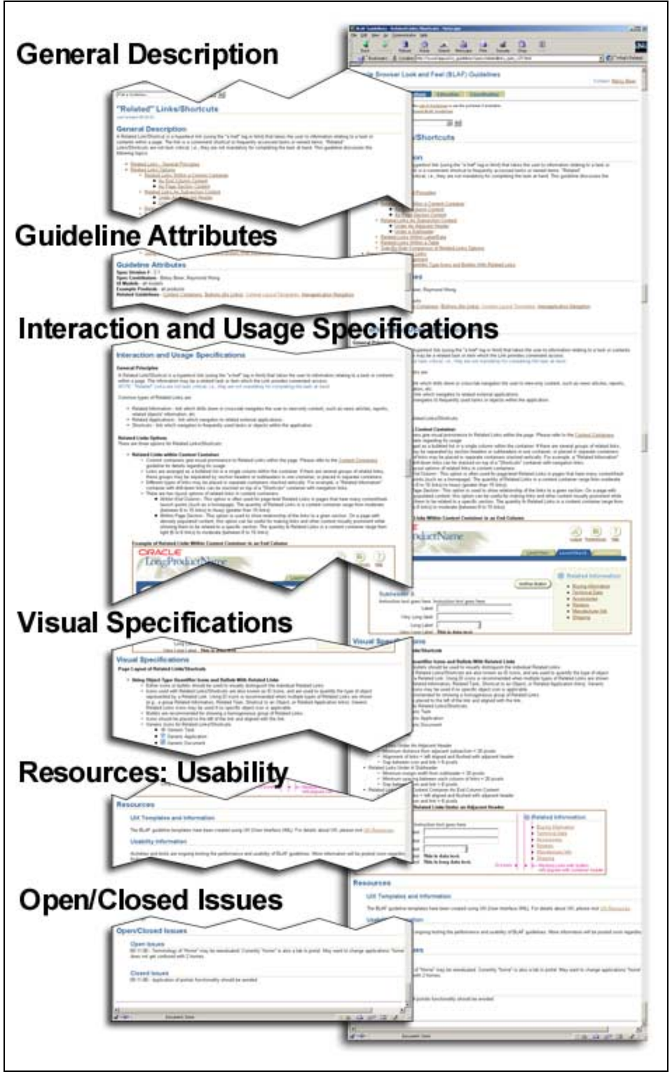
- A general description of the guideline.
- Guideline attributes, including a contact person from the UI group, a list of contributors, version number, products or product families using the guideline, and links to related guidelines.
- Interaction and usage scenarios for the guideline, including general principles of use, options for the given component/template/etc., and page flows indicating how the component/template relates to other parts of the guidelines.
- Visual specifications for the guideline that detail the color, size, minimum/maximum values, etc., and provide visual examples of the options for this component/template/flow.
- Usability data where test results validate components, templates, or flows. These tests may be product specific tests, or guidelines-wide tests. Usability tests

are ongoing and their data are incorporated into each guideline as results are available.

- Open and closed issues pertaining to the guideline.

Earlier we mentioned the value of tying the guidelines to UI code. A coding team meets with our lead guidelines designer in an on-going basis, and tracks the development of each component to an actual guideline.

Figure 5. Structure of a Guideline



VALIDITY OF THIS APPROACH

On its face, we believe the structure and content of the guidelines (in particular the multi-tiered nature of the guidelines) to be valid because they were developed with the assistance of UI designers, usability engineers, product managers, and development managers across the company.

The process involved each team sending a representative to a standing, weekly meeting to discuss new requirements with the lead guidelines developer. New requests would be examined to determine if they could be met using existing guidelines. If not, then they were tracked to determine how broadly they ranged across applications, i.e., one team's need for a UI widget might reflect a larger need for the component. If multiple instances of the need were apparent, then the lead guidelines designer would work with all impacted teams and UI designers to develop a solution. Several heuristics were used to assess and refine the solution:

- Was it compatible with or did it break the existing UI design guidelines?
- Was it scalable and extensible, i.e., could it handle very small sets of objects as well as very large sets of objects?
- Was it accessible to a screen reader?
- Could it be internationalized, e.g., did it handle 30% expansion for other languages and bi-directionality?
- Was it technically feasible from a coding perspective?
- Was it a usable design?

As each team indicated its requirements and participated in the design process, the multiple-levels of the guidelines took shape.

However, each team agreeing on the final output was not an indication that the designs in the guidelines were actually usable. To assess this more empirically, we conducted usability testing in both a top-down and bottom-up fashion, as another check on the guidelines validity. From the top down, we targeted specific guidelines for usability testing. These tests had to be generalizable to the entire product suite and range of users. To accomplish this we did the following: a) recruited three to four users per user type from across the different domains, b) developed tasks and content that were domain agnostic (e.g., working with human resources data or an employee directory), and c) developed our test plans, prototypes, and conducted our testing under constant peer-review. Various Page Components (e.g., page-level buttons), Page Templates (e.g., table personalization), and Page Flows (e.g., the save model) have been subjected to this kind of guideline-targeted usability testing.

From the bottom-up, we monitored our on-going product usability testing for guidelines issues. If we identified a potential usability issue with a guideline, then that issue was flagged by the usability engineer and communicated outward. Other engineers with products that had implemented similar components, pages, or flows, then determined if this was a recurrent issue. A guideline-specific test could be called for to design and test a new solution.

Finally, we can assess the validity of our approach by the degree of buy-in from upper management, and the degree of adoption by product teams. In December of 1998, Oracle had products with no common look-and-feel. In February of 2000, upper management called for use of the guidelines for all Web-based applications. By December of 2000, teams across all Web-based products were using the guidelines; over 100 Web-based applications have been implemented using the guidelines. In addition, the guidelines were published on the Oracle Technology

Network (http://otn.oracle.com/tech/blaf/) so that customers could build matching custom applications that integrate with Oracle applications.

LESSONS LEARNED

One challenge in growing the guidelines has been the tension between maintaining consistency across applications versus designing for each applications' unique needs. We address these requests by requiring that teams provide validation (e.g., documentation, usability data, user requirements data) that their needs are indeed unique.

Another challenge was the sheer task of maintaining the guidelines documentation. With each modification of the guidelines, a ripple of changes must be made throughout the documents. What began as a one-person project, has now grown to include three full-time personnel, plus cross-group collaborators. This team exists as a Standards group within (and is funded by) Usability and Interface Design.

A third challenge has been incorporating the requirements of external groups (e.g., system performance, accessibility compliance, and cross-browser compatibility) into the guidelines. To develop guidelines that are responsive to these external forces, we recruited allies throughout the organization with special knowledge in each of these areas to provide consultation and insight.

CONCLUSION

The Bull's-Eye framework and the resulting set of guidelines are a step forward from where we started. The levels of the Bull's-Eye provide a framework for thinking about how all the types of guidelines work together, from the general principles to the specific components. With this framework we bridged the gap left by other guidelines and standards, as well as designed for a wide range of users and application domains.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers of this paper, as well as, Dr. Elizabeth Boling, Lawrence Najjar, Michelle Bacigalupi, Daniel Rosenberg, Dr. Anna Wichansky, and Irene Wong for feedback and comments. In addition, this work would not be possible without the often invisible, but always invaluable, hard labors of our fellow user interface designers, usability engineers, product managers, and developers at Oracle.

REFERENCES

1. Apple Computers, Inc. *Macintosh Human Interface Guidelines*. Addison-Wesley, Reading, MA, 1992.
2. Gale, S. A collaborative approach to developing style guides, in *Proceedings of CHI '96* (Vancouver Canada, April 1996), ACM Press, 362-367.
3. Henninger, S. Haynes, K., Reith, M. W. A framework for developing experience-based usability guidelines, in *Proceedings of DIS '95* (Ann Arbor MI, August 1995), ACM Press, 43-53.
4. IBM Corporation, *Object-Oriented Interface Design: IBM Common User Access Guidelines*. QUE, Carmel, IN, 1992.
5. McFarland, A. & Dayton, T. *Design Guidelines for Multiplatform Graphical User Interfaces* [Document # LP-R13]. Bellcore, Piscataway, NJ, 1995. Available at: http://telecom-info.telcordia.com/site-cgi/ido/index.html
6. Microsoft Corporation, *The Windows Interface Guidelines for Software Design*. Microsoft Press, Redmond, WA, 1995.
7. Miller, A. Integrating human factors in customer support systems development using a multi-level organisational approach, in *Proceedings of CHI '96* (Vancouver Canada, April 1996), ACM Press, 368-375.
8. Najjar, L. J. E-commerce user interface design for the Web, in M. J. Smith, G. Salvendy, D. Harris, & R. J. Koubek, *Usability evaluation and interface design: Cognitive engineering, intelligent agents, and virtual reality*, Vol. 1, Erlbaum, Mahwah, NJ, 2001, 843-847.
9. Ohnemus, K. R. Web style guides: Who, What, Where, in *Proceedings of SIGDOC '97* (Snowbird UT, Month 1997), ACM Press, 189-197.
10. Oracle Corporation. *Oracle Applications User Interface Standards Release 11*. [Part # A58193-01]. Redwood Shores, CA, 1998.
11. Roszenweig, E. Design guidelines for software products: A common look and feel or fantasy? *Interactions*, *3*(5), (September/October 1996), 21-26.
12. Sun Microsystems, Inc. *Java Look and Feel Design Guidelines* [2nd ed]. Palo Alto, CA, 2001. Available at: http://java.sun.com/products/jlf/
13. Sun Microsystems, Inc. *Java Look and Feel Design Guidelines: Advanced Topics* [2nd ed]. Palo Alto, CA, 2001. Available at: http://java.sun.com/products/jlf/
14. Tidwell, J. *UI Patterns and Techniques: About Patterns.* May, 2002. Available at: http://time-tripper.com/uipatterns/about-patterns.html
15. van Duyne, D. K., Landay, J. A., & Hong, J. I. *The Design of Sites.* Addison Wesely, 2002.
16. Weinshenk, S. & Yeo, S. C. *Guidelines for Enterprise-Wide GUI Design.* John Wiley & Sons, New York, NY, 1995.
17. Zhu, W. Designing and evaluating a Web-based collaboration application: A case study, in M. J. Smith, G. Salvendy, D. Harris, & R. J. Koubek, *Usability evaluation and interface design: Cognitive engineering, intelligent agents, and virtual reality*, Vol. 1, Erlbaum, Mahwah, NJ, 2001, 838-842.