Oracle Fusion Applications User Experience
Design Patterns: Productivity Realized
Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Executive Overview
Oracle Fusion Applications User Experience (UX) Design Patterns are a set of 150 reusable models of user interaction and flow that are being applied across Oracle Fusion Applications. The usage of these patterns has led to enhanced consistency, productivity and usability among our end users and developers. We spent thousands of hours studying users at their desks and in our usability labs to understand the common processes they follow and the optimal solutions to support them. The patterns are fully buildable with Oracle Fusion Middleware and are available to customers so that they can understand and use them when customizing or extending Oracle Fusion Applications. This paper describes the nature of a user experience pattern and tells how the patterns were used to build Oracle Fusion Applications.

Introduction
At the start of the Fusion Applications design cycle, the Oracle Applications User Experience (UX) team was in the unique position of being able to bring together the staff, expertise and user interface design materials from three companies: Oracle, PeopleSoft, and JD Edwards. Over time, the acquisition of Seibel and other companies continued to impact the process. We had three key goals:

• Design user experiences that supported productivity, learn-ability, and consistency

• Provide a structure that enabled an efficient design and development life cycle

• Enable our customers to make use of our tools and methods when they customize or extend our products

Creating a library of UX design patterns represented the optimal approach to solving all of these challenges. We were able to bring together the existing patterns from the then-current product lines and further develop many more that targeted Oracle Fusion’s unique needs. We discovered, usability tested, and documented the Fusion User Experience Design Patterns early in the design process and then used them across the design of Oracle Fusion Applications. The reuse of these tested, proven patterns has
It is often said that no group of people can create a work of art, or anything which is whole, since different people pull in different directions, and make the end product a compromise which has no strength... The use of a shared pattern language solves these problems... a group of people who use a common pattern language can make a design together just as well as a single person can within his mind.

Christopher Alexander, The Timeless Way of Building

contributed to a user experience that is more efficient and has a consistent look and feel across the suite. The patterns were designed to be easily understood and applied by a range of users: product managers, developers and user experience professionals.

For each pattern, we partnered with Fusion Middleware development to provide reusable components, sample code and worked examples to support a smooth transition from design to development. For the most common patterns (pages and tables among them), new components were created that were tightly coupled to the user experience patterns. This led to increased code reuse and higher developer productivity.

Ultimately, the Oracle Fusion User Experience Design Patterns became a key part of the Oracle Fusion Applications development process in both the design and coding stages.

The Design Patterns Approach

What is a Design Pattern?

A design pattern is a reusable solution to a common problem. The concept of design patterns was first developed by the architect Christopher Alexander, as part of his research into how people could be more empowered to build their own structures. Alexander’s thinking had a significant effect on computer science and programming in the 1960’s and 1970’s, and user experience design, beginning in the late 1990’s.

An example from Alexander’s pattern library is Entrance Transitions, which tells how you should design the way a person goes from outside a house to inside a house, including the path, the steps, the door, and other elements.

Patterns are related to each other. For example, Entrance Transitions has related patterns, including Building Fronts, Paths and Goals, and Entrance Room. The relationships among the patterns form a language from which you can build any kind of environment you require.
Oracle Fusion User Experience Design Patterns work in the same way. Each pattern describes a user problem or goal, the context in which the pattern applies, and one or more solutions that solve the problem. Each pattern also references related patterns, in this way forming a design language that can be used to provide a complete solution across multiple user experience problems. The user experience of Oracle Fusion Applications is one such complete solution.

Studying Our Users

Patterns are discovered through observing the processes users follow to accomplish their tasks and understanding what works and does not work for them. We sent our 130-person Applications UX organization out to companies around the world. Through observations taking over 1,500 hours, we gained an in-depth understanding of our users and how they work.

Collecting Possible Patterns

We reviewed patterns that had already been discovered and used in the current versions of PeopleSoft, JD Edwards and Oracle EBS applications. We selected the best of these patterns to move forward into Fusion.

We also did an extensive analysis of Web 2.0 patterns, including social media and rich application interfaces. We analyzed pattern books, pattern websites and consumer applications to find patterns relevant for enterprise software.

Finally, as part of our re-envisioning of user experience for enterprise applications, we generated new patterns to support modern, Web 2.0 user experience models. UX designers, product managers, developers and documentation writers all participated in the creation of these patterns.

Testing the Patterns’ Usability

All candidate patterns were tested in both stand alone and application contexts to ensure that they were usable and desired by users. We created prototypes that hooked the patterns together into flows that modeled real world tasks. Actual enterprise software end users, such as line managers, accountants, sales representatives and dozens of others, worked through scenarios that tested the patterns. This testing was done early to ensure that what was built for Fusion works for users.

Creating a Pattern Library

Our combined research, design, and testing produced a library of over 150 user experience design patterns. These patterns range from traditional step by step “train” models to advanced Web 2.0 interactions like embedded chat and contextual actions. Some patterns are rather simple: adding an attachment, for example. Other patterns are complex: master detail layouts, or hierarchical displays. Some patterns are single entities; others are grouped in a set of related
patterns, with guidance for how to choose the correct one from the set. Each pattern has multiple, known usages.

The Pattern Template

In the library, each pattern is presented in a standardized template. This template was carefully designed. Early in our research, we interviewed and studied product managers and developers, two key audiences for the patterns. We learned that they wanted to be able to understand a pattern quickly, preferably through illustrations. So our pattern template emphasizes using illustrations, screenshots, tables and cross linking, rather than using a lot of text.

Figure 1 shows a pattern from the library: Hover Details. This pattern is part of the pattern set called Detail on Demand. On the left, the full library of patterns is always available for quick navigation. The pattern template itself consists of:

- A brief, easy to scan description of the pattern (1)
- The pattern set decision table, which shows all the patterns in a set, and the factors you use to choose a pattern (2)
- One or more sample designs or screenshots (depending on the complexity of the chosen pattern), showing how the pattern could be used (3)
- A table showing the required and optional screen elements for the pattern (4)
- Links to related patterns and guidelines, the developer guide, and any change history (5)
Figure 1. Oracle Fusion Applications User Experience Design Pattern example
Making Patterns Easy to Find

With a library this large, designers risk not finding all possible patterns. Our library provides a couple of ways to find patterns. First is a traditional site level keyword search. Second is a pattern filtering tool that we invented, called Design Filter Tool (DeFT).

DeFT enables you to filter the complete set of patterns by selecting criteria that the pattern should meet. You can select the end-user type (e.g., experienced, casual) and the core task type (e.g., create, monitor, search) and the tool filters the pattern list with matching results. This not only speeds pattern finding, but also keeps the focus on the problem and the user. It also provides an educational benefit: as designers search for possible patterns, they learn about the many options available. The DeFT tool is available with the Oracle Fusion User Experience Design Patterns.

![DeFT Tool](image)

Figure 2. The Design Filter Tool enables you to view user experience design patterns that match particular criteria.
Making User Experience Patterns Easy to Code

One of the most consistent pieces of feedback we hear from developers and business analysts is that design patterns need to be completely buildable with the development technology and platform. To support this, we ensured that every pattern was entirely buildable with Oracle Fusion Middleware. This includes technologies from Oracle Application Development Framework (ADF), Oracle Business Intelligence Enterprise Edition, and other coding tools. Additionally, we created a set of higher level “application components” that encompass and launch many of the patterns, for example, the UI shell component, in which all Oracle Fusion Applications run. This componentization approach effectively bridges the UX patterns and the underlying code behind every Oracle Fusion page.

We also made it easy to find the code support for each pattern, so that developers could immediately see what was involved in coding a particular pattern. On each pattern there is a link to the pattern development guide (see Figure 3). This development guide shows how you can either use an applications component or build the pattern out of base ADF components. This helps developers close the loop easily between design and development.

Growing the Pattern Library.

Unlike a book of patterns, an online pattern library can and should be a living document. As Applications technology changes and more requirements and use cases (a use case is a description of the steps a user takes to accomplish a task) are turned into flows, more patterns are identified and added to the library. As Oracle Fusion Applications continue to grow, patterns will continue to be developed as we design more applications and watch emerging industry trends.
A Design Patterns Usage Example: Create Invoice Work Flow

The Create Invoice work flow in Oracle Fusion Financials Accounts Payables is a great example of how design patterns were used in Fusion Applications.

Understanding Users, Tasks and Environment.

The first step in any user experience design process is to understand the users of the design: what goals do they have, what they must do in the work flow, and what environment or context they work in. To establish this understanding for Create Invoice, we conducted a series of customer site visits where we observed AP clerks doing their jobs. We learned that the clerks were highly trained users who were focused on getting their jobs done as efficiently as possible. They were “heads down” users, who preferred the keyboard over the mouse for efficiency’s sake. They were expert users of the application, who often knew every short cut key available. We knew that design patterns that supported efficiency and speed would be important.

Wire Frame Modeling with Patterns

Just as the design of a house begins with sketches and blueprints, the design of a user experience begin with wire framing. Wire framing is the process of sketching out the interaction and work flow quickly at a low level of fidelity, in order to explore many possible solutions. Here is where Oracle Fusion Applications UX Design Patterns are first applied. We explored different models as we worked to get the key elements of the design correct. In Figure 4, you can see where whole sections of the design requirements are met by patterns: Global Area (1), Task Pane (2), Quick Create (3), Quick Search (4), Create in a Form (5), Create in a Table (6), and Contextual Area (7). By picking patterns that aligned with the user goals of productivity and efficiency, the designers were able to come to a solution much more quickly, since they didn’t have to design each part of the page from the ground up. Some examples of how these patterns met the design needs:

- The Global Area pattern provides the user with continuous access to tools that apply across all flows.
- The Create in Table pattern, with pull down menus, allows users an efficient spreadsheet-like interface for adding lines.
- The Contextual Area pattern provides in-place information about data or conditions that the user may need to refer to while doing this task.
Interactive Prototyping and Validation with Users

Once the wireframe was complete, we created an interactive prototype that we could test for usability with end users. Our prototyping tool, based in a third-party web page design tool, contains code snippets and templates for many of the design patterns, making the prototyping easier and faster. With a working prototype we were able to conduct usability evaluations on the flow with customers well in advance of user interface coding. We tested the flow with 14 participants from six customers. We used this feedback to re-inform and improve the flow, and ultimately, the patterns themselves.

As an additional validation of usability, we conducted a productivity analysis to see whether the coded flow would result in fewer steps to complete a task than the existing application. The new design for the Create Invoice workflow is estimated to provide a 27% improvement in productivity over the existing application, based on keystroke-level modeling. Analysis of the new vs. old work flows show that the consistency and efficiency created by these usable patterns may lead to the productivity improvement.
Coding the Flow

The extensive use of user experience patterns in the Create Invoice flow meant that ADF code components could be reused during development, saving development time and ensuring consistency across work flows. Figure 5 shows a typical page where reusable components were used. From top to bottom:

- The UI shell component (1) comprises the top, left and right areas of the page. It provides access to a consistent set of navigation functions such as the Main Menu, Favorites, Global Search, Preferences and other items, as well as tasks, quick actions, and contextual information.

- Next, the Applications Panel component (2) provides a consistent set of navigation buttons and an easy and consistent way to include headers and footers across all pages.

- Finally, the Applications Table component (3) provides the core table functionality and also the ability to launch the attachments pattern.

Figure 5. Screen shot of the coded Oracle Fusion Create Invoice work flow, showing overlays that indicate where ADF code components were used.
This combination of the UI Shell, applications panel and applications table components automatically took care of:

- Positioning and labeling of branding, global search, preferences, help and logout functions
- Page button order and naming
- Page header location and style
- Table menu default actions
- Launch point and wiring for create, update and delete functions

Avoiding manual work not only saved development time, but it made it easier to code a user experience that is consistent with other flows across the Fusion Applications suite.

Conclusion

Oracle has made design patterns a key part of the design and development process for Oracle Fusion. We built an extensive library of patterns that have a strong connection with Oracle Fusion Middleware. These patterns were leveraged extensively across the Oracle Fusion Applications suite to drive user experience consistency and may result in higher user productivity. Developers found it easy to build pages based on patterns due to the strong connection between patterns and Fusion Middleware code. The success of the Oracle Fusion UX Design Patterns has led to us to create new pattern libraries dedicated to Mobile Applications and User Assistance (embedded help, messages, etc.).

From here, the patterns story will continue. We are focused on ensuring that our customers will be able to leverage our patterns when they are making their own customizations and extensions to Oracle Fusion, through externalization of our patterns. We understand that as you work to provide your users and customers with flows that are productive, knowing how Oracle Fusion Applications were designed and built will contribute to your ability to accomplish this successfully.