Using Patterns to Capture Design Experience

By James Hobart, President, Classic System Solutions
December 2001

Design guidelines have been used with some success to capture design knowledge and to help developers and designers make the best decisions when creating a user interface. We have long been a fan of this approach and have provided the software industry with a very robust set of design guidelines integrated into a web-based knowledge portal to help improve software interface design and avoid making the same design mistakes over and over again. Despite all this effort, applying guidelines on your individual projects still has its challenges. Because the growing complexity with interface design, the guidelines needed to be adhered to and followed are numerous and it is often difficult to select the right set of guidelines that apply best in a particular situation. A few key people within you organization may have acquired these skills as part of a usability team, however this knowledge is rarely held among the vast majority of your design and development staff. The reality is they are often too busy writing code and testing for bugs!

To address this issue, we believe the next evolution in capturing and implementing design knowledge will be with the use of visual design patterns. **A design pattern is a structured textual and graphical description of a proven solution to a recurring design problem.** Patterns offer a powerful new way of focusing on design solutions based on specific context by telling the designer when, why and how the solution can be applied successfully. Therefore, we feel patterns can be a very powerful way to leverage your existing guidelines as tools for developers. An important goal of any design team is to capture the reasons for design decisions, and the experience from past projects. In essence, to create a corporate memory of design knowledge which can be easily accessed and contributed to while your projects are being developed.

Effectively capturing design solutions:

- Ensures the best solutions of the past are captured and implemented on future projects.
- Helps to avoid repeating design errors from previous projects.
- Can introduce new team members to the design decisions of past projects.
- Can be used to train and educate less experienced designers or team members to the best practices of user interface design within your company.

A strong business case can be made to invest in capturing visual design patterns. Departing employees often take most of the memory and experience from their projects with them, and the enterprise cannot refer to that knowledge anymore to handle similar design problems in subsequent projects more effectively.

How to implement design patterns

Successfully documenting and implementing patterns is actually quite difficult. A solutions framework using design patterns must be developed based on best practices for your deployment platforms and for your specific vertical industry needs. High level design patterns must be created to address large-scale design issues and reference lower-level patterns to describe their solution. For example, an advanced search pattern will likely reference lower level patterns like results list pattern and simple search pattern.

So, if you are going to adopt this approach, what makes an effective pattern?

Patterns must reflect real world examples

Patterns are more effective when supported with guidelines and real-world examples.
Get your design team to real examples where experiential learning can take place and the designers and developers can play with and test the potential solutions in the context of their specific situations. Support the patterns with proven guidelines, case studies, examples of known usage, checklists and code examples to bring them to life. Constantly validate and improve the patterns as a result of real world user experience and usability testing. If you have a usability team, this approach highly leverages your existing team experience among the entire corporate development community.

Don’t make patterns too abstract

Many existing patterns that have been documented in the software engineering world are written at too abstract of a level to be easily understood and embraced by users of a typical design team. This is an area we differ from others in the design world. Some pattern proponents believe in keeping patterns at a high-level to ensure their ‘timeless’ nature. In theory this sounds good, however developers ultimately are looking for coded examples of a pattern. The closer you can come to showing them a solution for their specific implementation, providing examples and then linking them to actual code, the better chance you will have at successfully implementing this approach.

Use a common pattern language

A common approach to documenting patterns will be emerging in the coming years using a Common Pattern Language. This will allow a broader range of people within your organization to consume, contribute and create new patterns ultimately growing your design knowledge base. Currently, the most common language has been based on work by architect Christopher Alexander and his book, The Timeless Way of Building.

We have been creating visual design patterns for clients now for over two years and are excited to now be including these solutions in our consulting practice, training and design knowledge products. Give us a call or email me jimh@classicsys.com and we’ll give you a sneak preview of our next release of GUIguide, our design knowledge portal which is now centered around this ‘Solutions Driven’ approach to capturing the knowledge and experience of how to design more usable software.

About the author:

James Hobart is an internationally recognized user interface design consultant based in California, USA. He specializes in the design and development of large-scale, high-volume client/server and web applications. He is an expert in GUI design for transaction processing systems and strategies for migration to thin-client graphical user interfaces. He can be reached at jimh@classicsys.com

Find more articles on Usability at www.classicsys.com