

Book Review:

PHP Objects, Patterns, and Practice

Introduction

Everyone by now has heard of Patterns. Many people have found easy opportunities to use them in their day-to-day work. Others feel that they are not using them properly or they do not have the opportunities. In fact there is a pattern developing of those that buy-in to this new language and those that don't. For Patterns are a new language, and the basic language of Patterns is the UML.

Now you do not need to be a UML expert to understand patterns. All you need to be able to read and understand are Class diagrams which cover Interfaces, Abstracts and the various Associations.

Let's be clear that patterns existed long before the topic of patterns was recognised. Every programmer who was worth their salary had standard approaches to certain categories of problem. These patterns of standardised approaches were usually specific to the implementation language, and this limited their domain of application. The difference nowadays is that the patterns can be expressed in UML (not forgetting the complementary English text in this) and since UML assumes only OO capabilities in an implementation, patterns expressed in UML are inherently portable.

Once we get the vocabulary of patterns under our belts then we, as programmers, designers and testers, can start working at a higher level. No more arm waving and explaining some complex call sequence, we can just name the pattern we are using.

OK, so given the advantages, why aren't patterns in more use? IMHO, one of the reasons is the excessive formalism of some of the descriptions of the Patterns themselves. If you think of Patterns as a toolbox of solutions then they will not help if you do not understand or accept the problem! Moreover, in an attempt to be general, many patterns are overkill for simpler problems and there is a real cost to using them. Standard they may be but if they are excessively complex they may be inappropriate. I am sure that this last comment will not stand for long. As Patterns of all sorts become mainstream and even embedded into programming languages (rather than design tools) it will be easier to use them, regardless of their innate complexity, because they will (become) tried and trusted solutions.

So, what's all this got to do with the book? Well, this book is one which approaches patterns from a consistent design perspective and takes into account the possible overkill of use. Also, it provides pragmatic views of pattern use. For more details read on....

What's in the book?

The book runs to nearly 500 pages, in soft cover. There are six sections

Introduction

Covers PHP Design and Management

Running to just a few pages, this section introduces the subjects and some of the history.

Objects

Explains PHP and Objects, eventually addressing all of the core OO constructs. This section alone makes the book worthwhile. It is informative, discusses styles of use of OO in a clear and practical manner. All of the examples are given in PHP, which is probably not the first language of choice for major OO implementations. However, if you can get over any PHP prejudice, it is just another OO language, and with its OO extensions, should be quite readable for any programmer.

Patterns

This is the meat of the book. It discusses the formats of Pattern documentation and, very importantly, identifies the core principles of patterns. Since these principles will be familiar to any experienced programmer it should give them a warm feeling towards patterns. Patternitis - a disease you may not yet have, but may acquire is also mentioned - basically the unnecessary use of patterns. The range of subjects is large, covering basic object creation (Factories and the like), to flexible programming techniques (Decorator, Façade, Composition), task-oriented patterns (for example, Observer, Strategy, Visitor, etc), Database patterns(this is where the outside world get to talk to us via Data Mapper pattern, Identity object, Data Layer, and others) and finally the Enterprise patterns, addressing architectural patterns, something necessary for programmers coming from the traditional areas of PHP application.

Practice

This section looks at practicalities of large-scale development in PHP. The PEAR package is described in some depth and a lot of time is given over to the principles of Version Control, documentation and testing in general, combined with detail discussions of phpDocumentor, CVS, PHPUnit and Phing. Necessarily this is not going to excite the Java programmers (or others) amongst us, but the principles are clear and well described and there is enough overlap between the Java / .net toolsets to make this relevant.

Conclusion

The conclusion brings it all together in a good, concise summary of the previous sections. Experienced readers might actually like to start here, to whet their appetites for the rest of the book.

Appendices

Some good references are given and a complete PHP version of a Parser, based on Stephen Metsker's book *Building Parsers with Java*

Summary

The obvious negative for a general audience is the focus on PHP. Allowing for that, the book is very informative, not a high-brow view of Patterns at all and likely to be of great use to programmers who wonder why they are not buying in to patterns in their organisations. The book's style is easy to read and the examples given are practical and straightforwardly applicable. Highly recommended. PHP or not!

About the Reviewer

George Brooke is a Technical Trainer / Consultant as well as being a PRINCE2 Approved Trainer. As Head of Software Development at Siemens Nixdorf he managed solutions development for the Financial, Retail, Hotel and Energy markets for UK/European departments and customers worldwide, from concept through to delivery and roll-out. He runs a training and consultancy company based in Cambridge UK focusing on OO design with UML, programming and implementation techniques and IT project management . Contact him at [George Brooke](#)

Book Information

PHP Objects, patterns and Practice, Second Edition

www.apress.com

Matt Zandstra

ISBN-13: 978-1-59059-909-9

ISBN-10: 1-59059-909-8