JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2003**

Vol. 2, No. 5, September-October 2003

Book Review

Patterns of Enterprise Application Development

by Martin Fowler, Addison-Wesley, Boston, MA, 2003. 533 pp. $49.99(hardback). ISBN 0-321-12742-0

Reviewed by Charles Ashbacher

Fowler avoids giving a precise definition of an enterprise application, preferring to list a set of characteristics that most share. In general, they are very large systems, with many user interface screens used to concurrently access and update an enormous amount of data. In nearly all cases, the data must be persistent, in fact it most often is very persistent, meaning that it has to live through iterations of the software, alterations of the operating system, changes in the hardware, and staff and programmer turnover.

Furthermore, enterprise applications usually must communicate with other applications, which are often just as large and complex. Examples include payroll and patient records, credit card processing, insurance claim processing, banking, and foreign exchange trading. In short, most of the programs that run the modern global economy, which are many of the most complex software projects currently in use. Finally, the programs must be constructed so that they can be "easily and quickly" changed by people who did not create them to adapt to conditions that can change very quickly and often without any input from the programmer. With so much at stake, there must be a set of best practices, which is what is captured in this book.

The patterns of software construction explained by Fowler are generally in the small, in the sense that they describe specific operations rather than demonstrate a large architectural form. Each of the specific patterns is presented by first listing a one-sentence description of the purpose of the pattern and a UML diagram illustrating the structure. This is followed by sections describing how the pattern works, when to use it and one or more examples demonstrating specific implementations of the pattern using source code skeletons. Both C# and Java are used in the demonstrations, which does not create an understandability problem. The languages and contexts are so similar that anyone who can understand either one will have no problem reading and understanding the code.

Some examples of the fifty one patterns listed on the inside front cover are:

- Lazy load – where an object will load only the data currently needed, but does maintain links to all other data that may be needed.
- Front controller – a single handler object that consolidates all requests made for a web site. It can then send requests to the specific objects for services such as security, internationalization issues and specific displays targeted for particular users and locations.
- Optimistic offline lock – used to prevent conflicts when concurrent business transactions are executing. The solution is to roll back the transaction when a conflict is detected.
- Server session state – keeps the data for the session stored on a server in a serialized form.

While the examples are often of necessity extremely simple, they do illustrate some of the most effective and tested solutions to common software development problems. Therefore, this is a book that no builder of software that can be considered an enterprise application should be without. It is hard to believe that there is an enterprise application being constructed anywhere that does not involve the solving of many of the problems listed in this book.

My only complaint is the occasional bad English that appears. For example, on page 100 there is the phrase, "The only reason I've concentrating on Java . . . " and on page 119 the phrase "One factor that comes into this is comfortable used the development team is . . . " appears. While no book is error free, this type of error is frequent enough to make one wonder about the quality of the final editing.

There is nothing harder than making effective and efficient software that will run the IT equivalent of forever. That is what enterprise applications are supposed to do and if you are one of the minions tasked with doing your part to build one, then put yourself on the right path and read this book. You and everyone else who interacts with the software will be rewarded with a better experience.