
Introduction

Design patterns have been around for quite a few years. They were originally created by the Gang of Four (Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides), which is responsible for formalizing the accepted design patterns we use today. Their use, while questioned and argued over by different programming schools of thought, has generally been accepted as “best practices” within the development community. The original takeoff of design patterns was brought about in the Java and C++ world.

Other languages like .NET (dot NET) have sprung up in recent years to fill certain marketing and technology voids left open by these languages, and have adopted design patterns in one way or another to mimic what was learned by programmers in Java and C++. Even the previously script-oriented syntax of Visual Basic has blossomed into a fully object-oriented language, via Visual Basic.NET. Microsoft also added a Java-like language, which has related syntax and framework structure, in the form of C#, which is so much like Java that applying design patterns is a relatively expected conjecture.

Most programmers new to object-oriented languages seem to have the same dilemma when faced with whether or not to use or learn design patterns: *Why should I?* Usually the problem is that the ability and time required for learning patterns-based programming is at a premium and the return seems dubious. Someone not familiar with the why

behind the methodology certainly will not continue on to the what and where. So explaining to people new to the concept why they should spend the time learning design patterns can be difficult.

Another issue is availability and ease of dissemination of the relevant data. There have been numerous books and papers written on the uses and methodology of design patterns, but to a junior developer looking to improve basic skills or learn the skills needed to get a job, having to read complicated texts to learn a new methodology or way of thinking is out of the question. Senior developers looking to upgrade current skills can have similar issues, in that to augment their skill sets they have to do a large amount of research just to be able to understand the basics. Most of the texts available on design patterns are not easily referenced, and require an expanded understanding of object-oriented methodologies and patterns to even get through, and full comprehension may require reading the material several times from start to finish. There have been a few attempts to “dumb down” the data, but these seem to be more playful than useful, and leave project developers a little turned off by the lack of serious content of the work.

As a developer, as well as a development manager, I can assure you I have had the same problems. Whether learning to use patterns or teaching them to others, I found there was no quick and easy reference to explain or learn which pattern is appropriate in which case and why. I often struggled with the basics of object-oriented design when learning patterns, and had to learn the hard way which methods worked where. One thing I did find was that design patterns were created and have evolved from the work of everyday code practices, and many developers are

using them without actually knowing the established 23 patterns. When I first started to pattern, I often had to search through many texts and websites, accessing several sources to understand a particular pattern. This was hard when trying to meet deadlines, since time was crucial, and the only reason I persisted in learning was because I saw there was a lot of value in what I did manage to understand.

After several years of this kind of thing, I began to write some articles for the Code Project site (www.codeproject.com). I started with aspects of things about which I saw value, but I thought were under-documented. I wrote in a fashion that I would like to read, something without a lot of complicated dialog that just stated problems and solutions in code and explanations of what I was doing and why. After a while I realized a lot of people seemed to respond to what I had written. That was when I decided that I should write this book.