Contents

Preface

This book is the fruit of about 15 years of thinking about IT application development. It started when I was working on IT architecture in the late 1990s. At that time I wrote a book called *IT Architecture and Middleware: Strategies for Building Large, Scalable Systems* (a second edition coauthored with Peter Bye in 2004 is still available). This was about the technology for building integrated applications and about how to make the applications scalable, highly available, and secure. Other people were thinking along similar lines at the time, and the kinds of solutions Peter and I were proposing came to be called Service Oriented Architectures (SOAs) because the basic idea was to have reusable services with which you can rapidly assemble new applications using integration technology. In spite of the advantages of SOA, which we thought were obvious, very little happened. IT managers liked the SOA story but didn't get around to implementing anything. Something was missing, and almost from the beginning I had the suspicion that the missing something was application development. In other words, we didn't have a good answer to the question, "How do you develop an SOA application?" Or perhaps the question is better expressed as, "I have a bunch of requirements; what do I have to do to ensure that I end up with an SOA solution rather than a stand-alone application?" Over the next few years I did less and less thinking about architecture and more and more thinking about application development.

I first did application programming in the late 1970s. Since that time I have mostly worked in the system and environmental software arena, fixing and designing. I spent a lot of time mending data management software and occasionally would be thrown a compiler bug or an operating system bug to fix. I have done a fair bit of designing and programming system software. Later on I worked on database design and repository design (I could give a long discourse on version control—strangely, very few people wanted to hear it). By the year 2000, I was experienced in many aspects of computer technology, but I hadn't done a lot of straightforward application design and programming, so I couldn't in all honesty go to the application developers and tell them they were doing it all wrong.

At that time, application development gurus showed little interest in architecture. Instead they were at war among themselves. In one corner was the "big design up front" (BDUF) crowd who promoted designs based on Unified Modeling Language (UML) modeling. The designs were structured, well documented, and full of quality