The continual evolution of object oriented technologies creates both opportunities and challenges. The chapters in this book were selected to represent a variety of perspectives concerning the present and future of this broad sub-field of software development. Practical considerations limited the size of this book to twelve chapters, which forced the omission of several important topics such as object oriented programming.

The first four chapters are presented in systems development life cycle order. The opening chapter, written by Edward Sim, examines object oriented analysis (OOA), reviews some of the fundamental concepts on which OOA is based and discusses the acceptance of this new technology. Recently, many new techniques and methodologies have been introduced to assist analysts and users in efforts to identify and specify system requirements. One of the newest approaches to be used in this effort to improve requirements analysis is the application of object oriented analysis. Proponents of object technologies argue that the use of objects facilitates communication and problem understanding because people naturally think about their environment in object oriented ways. The solution for improving analysis and the requirements produced by that analysis, they argue, is to adopt an object oriented approach to doing analysis. However, despite these claims, the use of OOA has not achieved the levels of adoption that other object oriented technologies (i.e., programming languages) have achieved.

In the next chapter James Nelson, Kay Nelson, Mehdi Ghods, and Holly Lee discuss the use of structured design techniques in an object oriented environment. Their research examines specific traditional structured methods for their contribution to traditional development team performance. The attitude of the team toward structured methods and the satisfaction of the team with training in structured methods are used as mediating variables in this examination. Correlation analysis and stepwise regression are used as analysis methodologies. The results of these analyses are then mapped to the object oriented environment.

Samuel Agyemang describes object oriented testing. Most researchers and practitioners seem to agree that object oriented testing is a challenge. The main reason for this view seems to revolve around the fact that the objects and the code are inseparable, and also because of inheritance. Nevertheless, object oriented systems, when successfully tested, leave a better maintained product than traditional non-object oriented software.
Jozsef Komlodi examines the technical and market viability of object database technology. Object databases represent a revolutionary new technology and provide a superior storage facility for complex data structures and types. They also enable close language binding and a unified development process. It is a mature technology with advanced database management and development features, and has several proven and robust deployment examples. Besides its current technical excellence, this technology is also demonstrating future potential through such emerging technologies as Java, Application servers, and XML—a markup meta-language for documents containing structured information. The past failure of object databases to proliferate the market was mainly due to unawareness, lack of skills, and the overwhelming existing investment in relational systems. These factors are changing and new technology adoption is accelerating, so object databases are looking forward to a slow but sure take off.

A second grouping of chapters illustrates one of the main benefits of object technology—reuse, along with one of the main challenges—what to do with legacy systems. Jane Fedorowicz and Denis Lee provide an overview of software reuse and object technology. They surveyed practitioners with extensive systems development experience to evaluate their experiences with object oriented tools and techniques. A related goal of the study was to focus on reuse—to determine what is being reused and by whom.

The chapter by Gretchen Irwin and Chamini Wasalathantry provides an empirical study of reuse of object models. The aim of their study was to explore the effect of a reusable example on the cognitive processes associated with object oriented modeling.

Cobo and Mauco discuss transforming legacy systems into object oriented. They discuss how the development of new architectures and the improvements in programming methods and languages have created a need to reverse engineer and reengineer existing program code in order to get as much value as possible from legacy systems, while exploiting the latest technology.

Gerald Cameron considers integration and migration issues associated with upgrading legacy applications. Conversion of a COBOL legacy application to an object oriented application requires a complete restructuring of the legacy application. Objects and their inheritance structure must be identified, data usage and data flow must be analyzed, and instructions must be allocated to objects. Dynamic Object Oriented Programming allows parts of an application design that are represented by objects to be modified dynamically. Integrating or migrating legacy applications with newer more advanced client/server architectures can be a very expensive and time-
consuming undertaking.

The final four chapters address some of the more complex issues associated with object technologies. Alex Podaras introduces distributed object systems. The purpose of his chapter is to provide a clear understanding of what distributed object oriented systems are, no matter how complex they may appear to be. It will be shown that, fundamentally, distributed object oriented systems must have two object oriented properties or characteristics: encapsulation (the ability to hide code from the user) and messages (the way objects communicate). Additionally, it will be shown that software components (objects) of the distributed object oriented systems must have certain inherent features. Aside from the two object oriented properties and the certain inherent features, any critical system must have the ability to keep its data in a consistent state. This is particularly important when concurrent transactions are executed.

David Patton provides a discussion of distributed object business engineering. His chapter presents a framework for architecting enterprise-wide object based information systems. These next-generation systems maximize information value throughout the enterprise, while reducing development time and effort throughout the system lifetime.

Luis Proano examines industry trends in order to make recommendations for training approaches for object technology skills. He offers some ideas regarding the current needs in the information technology industry in terms of object oriented technology skills and knowledge. He also analyzes factors like the lack of mainstream products and object standards influencing the development of skilled professionals in working with object databases.

Robert Gittins brings together two important trends by questioning the use of object oriented technology for business process reengineering. He asserts that although the potential for object oriented technology has information technology and business professionals extremely excited, the burgeoning field is undeniably immature and currently lacks the stability necessary to be considered mainstream or a reliable option for companies that are about to reengineer their business processes. Despite the growing popularity of object oriented technology, there are numerous issues that have contributed to its inability to firmly entrench itself and take over from the older, proven technologies. Object oriented technology’s image problem has created a highly difficult decision-making process for corporations about to embark on business process reengineering (BPR) projects. At this time, reengineering with object technologies is a significant risk for companies to make and those who have moved forward with object technologies have not, for the most part, seen the results that they were hoping for and their organizations are now suffering as a result of this decision.