References

[Alexander et al.]

Alexander, et al. *A Pattern Language*. Oxford, 1977.

An inspiration for many people in the patterns movement. I'm less enamored of it than most, but it's worth looking at to understand an approach that so many draw so much from.

[Alpert et al.]

Alpert, Brown and Woolf. *Design Patterns Smalltalk Companion*. Addison-Wesley, 1998.

Little known outside the Smalltalk community, this book expands and explains many of the classic patterns.

[Alur et al.]

Alur, Crupi, and Malks. *Core J2EE Patterns*: *Best Practices and Design Strategies*. Prentice Hall, 2001.

One of the new wave of patterns books that put new life into the form. Although the patterns are expressed specifically for the J2EE platform, most also make sense in other places.

[Ambler]

http://www.ambysoft.com/mappingObjects.html

A useful source of ideas on object-relational mapping.

[Beck XP 2000]

Beck, *Extreme Programming Explained*. Addison-Wesley, 2000.

The manifesto for Extreme Programming. It should be read by anyone interested in software process.

[Beck Patterns]

Beck. *Smalltalk Best Practice Patterns.* Prentice Hall, 1997.

Undeservedly little read because of its Smalltalk base. It has more good advice for other OO languages than most books that are specially written for them. The only downside is that you might realize how much we all miss by not programming in Smalltalk.

[Beck TDD]

Beck. *Test-Driven Development: By Example.* Addison-Wesley, 2003.

Due to be out on the same day as this book. TDD is Kent's guide to the tight cycle of testing and refactoring that can evolve a design.

[Bernstein and Newcomer]

Bernstein and Newcomer. *Principles of Transaction Processing.* Morgan Kaufmann, 1997.

An excellent introduction to the head-hurting world of transactions.

[Brown et al.]

Brown et al. *Enterprise Java Programming with IBM Websphere.* Addison-Wesley, 2001.

Although two-thirds of this book is a software manual, the other third packs more good design advice than do most entire books devoted to the subject.

[Brown and Whitenack]

http://members.aol.com/kgb1001001/Chasms.htm

One of the earliest, and best, papers on object-relational mapping.

[Cockburn UC]

Cockburn. *Writing Effective Use Cases.* Addison-Wesley, 2001.

By far the best reference on use cases.

[Cockburn PloP]

Cockburn, "Prioritizing Forces in Software Design," in [PLoPD 2].

A discussion of application boundaries.

[Coleman et al.]

Coleman, Arnold, and Bodoff. *Object-Oriented Development: The Fusion Method, Second Edition*. Prentice Hall, 2001.

Although much of this pre-UML book is primarily of historic interest, its discussion of the interface model is very helpful to those designing a service layer.

[Evans and Fowler]

http://martinfowler.com/apsupp/spec.pdf

A discussion of the Specification pattern.

[Evans]

Evans. *Domain Driven*. Addison Wesley, in preparation.

A book on developing domain models. Although I don't usually like to reference books not yet published, the manuscript promises a fascinating discussion of an important and difficult aspect of enterprise application developments.

[Fowler Temporal Patterns]

http://martinfowler.com/ap2/timeNarrative.html

Patterns dealing with object histories that change over time.

[Fowler AP]

Fowler. *Analysis Patterns*. Addison-Wesley, 1997.

Domain model patterns.

[Fowler Refactoring]

Fowler, *Refactoring*. Addison-Wesley, 1999.

A technique for improving the design of an existing code base.

[Fowler CI]

http://martinfowler.com/articles/continuousIntegration.html

An essay that explains how to automatically build software several times a day.

[Gang of Four]

Gamma, Helm, Johnson, and Vlissides. *Design Patterns*. Addison-Wesley, 1995.

The seminal book on patterns.

[Hay]

Hay. *Data Model Patterns*. Dorset House, 1995.

Patterns of conceptual models from a relational perspective.

[Jacobson et al.]

Jacobson et al. *Object-Oriented Software Engineering*. Addison-Wesley, 1992.

An early book on OO design; introduces use cases and the interface-controller-entity approach to design.

[Keller and Coldewey]

http://www.objectarchitects.de/ObjectArchitects/orpatterns/index.htm

An excellent resource for object-relational mapping.

[Kirtland]

Kirtland. *Designing Component-Based Applications*. Microsoft Press, 1998.

Description of the DNA architecture.

[Knight and Dai]

Knight and Dai. "Objects and the Web." *IEEE Software*, March/April 2002.

An excellent paper on model view controller, its evolution and use in Web applications.

[Larman]

Larman. *Applying UML and Patterns, Second Edition*. Prentice Hall, 2001.

Currently my first-choice introduction to OO design.

[Lea]

Lea. *Concurrent Programming in Java, Second Edition*. Addison-Wesley, 2000.

If you want to program with multiple threads, you need to understand this book first.

[Marinescu]

Marinescu. *EJB Design Patterns*. New York: John Wiley, 2002.

Recent patterns book for Java's EJB.

[Martin and Odell]

Martin and Odell. *Object Oriented Methods: A Foundation (UML Edition)*. Prentice Hall, 1998.

Object modeling from a conceptual perspective, as well as investigation into the foundations of what modeling is about.

[Nilsson]

Nilsson. *.NET Enterprise Design with Visual Basic .NET and SQL Server 2000*. Sams, 2002.

A solid book on architecture for the Microsoft platform.

[Peckish]

two million (see page 79)

[PLoPD 2]

Vlissides, Coplien, and Kerth (eds.). *Pattern Languages of Program Design 2*. Addison-Wesley, 1996.

Compendium of patterns papers.

[PLoPD 3]

Martin, Buschmann, and Rielhe (eds.). *Pattern Languages of Program Design 3*. Addison-Wesley, 1998.

Compendium of patterns papers.

[POSA]

Buschmann et al. *Pattern-Oriented Software Architecture*. Wiley, 2000.

The best book on broader architectural patterns.

[Riehle et al.]

Riehle, Siberski, Baumer, Megert, and Zullighoven. "Serializer," in [PLoPD 3].

In-depth description of serialization of object structures, particularly when you need to serialize into different forms.

[Schmidt]

Schmidt, Stal, Rohnert, and Buschmann. *Pattern-Oriented Software Architecture, Volume 2*. New York: John Wiley, 2000.

Patterns for concurrent and distributed systems. More for people who design application servers than for those who use application servers, but it's good to have some knowledge of these ideas when you use the results.

[Snodgrass]

Snodgrass. *Developing Time-Oriented Database Applications in SQL*. Morgan-Kaufmann, 1999.

How to deal with tracking historic information in relational databases.

[Struts]

http://jakarta.apache.org/struts/

A Web presentation framework for Java that's growing in popularity.

[Waldo et al.]

Waldo, Wyant, Wollrath, and Kendall. *A Note on Distributed Computing*. SMLI TR-94-29, *http://research.sun.com/technical-reports/1994/smli_tr-94-29.pdf*, Sun Microsystems, 1994.

A classic paper on why "transparent distributed objects" is a dangerous oxymoron.

[wiki]

http://c2.com/cgi/wiki

The original wiki web, developed by Ward Cunningham. A rambling but fascinating open Web site where all sorts of people share all sorts of ideas.

[Woolf]

Woolf. "Null Object," in [PLoPD 3].

A description of the Null Object pattern.

[Yoder]

http://www.joeyoder.com/Research/objectmappings

A good source of object-relational patterns.