# Design patterns in communications software [Guest Editorial]

**Article** *in* IEEE Communications Magazine · May 1999

DOI: 10.1109/MCOM.1999.755445 · Source: IEEE Xplore



**1 author:**

Linda Rising
Independent Researcher
**56** PUBLICATIONS **872** CITATIONS

# Design Patterns in Communications Software

Linda Rising

I'm hoping that all of you who are new to patterns will find this Feature Topic a useful introduction to a powerful means of capturing solutions to recurring problems. Patterns is one of the "hot" topics in software development today. I often recall the 1994 Object-Oriented Programming Systems Languages and Applications (OOPSLA) conference where an incredibly long line of software developers queued up at Addison-Wesley's booth, waiting to buy a copy of the new book Design Patterns: Elements of Reusable Object-Oriented Software [1]. Sales are still breaking records and more patterns books are being published (see http://hillside.net/patterns/books). The patterns community is growing.

My goal with this Feature Topic for IEEE Communications Magazine is to meet the needs of two groups of readers. First, for those of you who are new to this topic, the lead article, "Patterns: A Way to Reuse Expertise," will introduce the notion of patterns and perhaps provide those who know a little about patterns with some new information. For those who are experienced with patterns, and for those newcomers who have read the lead article, the remaining five articles share the authors' experience with patterns in the domain of communications software.

The article by Michael Duell, "Managing Change with Patterns" describes the use of Mediator, a popular pattern from the Design Patterns text [1]. Stories of experiences similar to our own are always compelling. It's a human characteristic that we are intensely curious about others who have problems like ours. Hearing successful solutions to recurring problems is the driving force behind the patterns literature. When we read a well-written pattern, we identify with the problem and context, we feel the forces pushing and pulling us in different directions, and we feel an enormous sense of resolution when we encounter the solution. A powerful way to learn lessons!

Gerard Meszaros' article, "Design Patterns in Telecommunications System Architecture," not only provides a valuable overview of telephony but also shows those of us who work in this industry some patterns that are familiar but perhaps have gone without names. This article points out another benefit of patterns, giving names to solutions we all know, so that we can have a common vocabulary to tackle problems in our domain.

We are all, both as developers and users, interested in Web-based applications. "OpenWebServer: An Adaptive Web Server Using Software Patterns," by Junichi Suzuki and Yoshikazu Yamamoto, shows us an application of a little-known architectural pattern, Reflection. I know that my understanding of this pattern increased considerably after reading the article. The authors also proposed an intriguing example for a recent OOPSLA workshop on real-world, non-software examples of architectural patterns. More information on their examples and others from this workshop can be seen at http://www.agcs.com/patterns/papers/index.htm.

Certainly middleware, CORBA, and COM are topics high on everyone's list and certainly Doug Schmidt and his colleagues, like Chris Cleeland, are the experts we turn to for information about this critical area. Their article, "Applying Patterns to Develop Extensible ORB Middleware," shows us a host of patterns for distributed systems and the impressive framework, TAO, developed by the authors and others at Washington University. We all know that building frameworks isn't easy, so there are valuable lessons for us in this report.

The last article is an unusual one. Unlike the other articles in this collection that relate experiences with using patterns, "An Overview of Selected Call Processing Patterns," by Greg Utas, outlines a group of related patterns that can be applied in the domain of call processing. Those who are familiar with this area will be able to identify these patterns with an "Aha!" and those who do not know the area can, in many cases, find solutions that have been used to solve problems in other, closely related domains. The wonderful thing about patterns is that we find them in specific domains and can then generalize them to apply to other areas.

I hope you enjoy this interesting collection of articles about patterns. There are references, including URLs at the end of my article for further information. I also hope that all of you feel a kinship with the patterns community after having read these articles. If you are interested in contributing to the growing literature of patterns for communications, see http://www.bell-labs.com/cgi-user/TelePlop/TelePlop?FrontPage for more information. I would also welcome comments and questions about this Feature Topic.

## ACKNOWLEDGMENTS

Thanks to Ram Batni at AG Communication Systems for proposing the idea of this Feature Topic on design patterns in communications software.

*Editorial Liaison: R. P. Batni.*

We have not only included articles about patterns experiences but we have also used a process from the patterns community. It's called shepherding. When articles are submitted for publication, they typically go through a review process. This process involves a collection of experts who read the submissions and pass judgement on the articles, accepting or rejecting them, in some cases, with very little feedback for the authors. Shepherding, on the other hand, involves a closer working relationship between reviewers and authors. In this relationship, shepherds work with the authors, giving feedback to improve the submission. I hope you will agree that the result is better articles. As a side benefit, authors and shepherds get to know one another and a community develops. Since each author knows the shepherds, they can be mentioned in the acknowledgements, so here they are — all the reviewers of these articles. We owe all of them our sincerest thanks for helping to make this a very special Feature Topic. Thanks!

Paul Bramble, AG Communication Systems
Bill Davies, AG Communication Systems
Dennis DeBruler, Lucent Technologies
David DeLano, AG Communication Systems
Jim Doble, Motorola
Michael Duell, AG Communication Systems
Jeff Garland, Motorola
John Goodsen, RADSoft
Bill Haney, AG Communication Systems
Robert Hanmer, Lucent Technologies
Neil Harrison, Lucent Technologies
John Letourneau, Lucent Technologies
Patrick McDaid, Dovetail Computing
Gerard Meszaros, Solution Frameworks
Bill Opdyke, Lucent Technologies
Greg Stymfal, AG Communication Systems
Roger Tomas, AG Communication Systems
Greg Utas, Nortel Networks
Lizette Velazquez, Lucent Technologies
Steve Vinoski, IONA Technologies

## REFERENCES

[1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Reading, MA: Addison-Wesley, 1995.

## BIOGRAPHY

LINDA RISING (risingl@agcs.com) is a member of the Communications Technology Center at AG Communication Systems. She has a Ph.D. from Arizona State University in the area of object-based design metrics. Her background includes university teaching experience as well as work in industry in the areas of telecommunications, avionics, and strategic weapons systems. She has been working with object technologies since 1983. She has contributed to a number of publications, including several recent ones on patterns, "A Training Experience with Patterns" in the October issue of *Communications of the ACM*, "Patterns: Spreading the Word" in the December 1996 issue of *Object* magazine, "The Road, Christopher Alexander and Good Design" in the March 1997 issue of *Object* magazine, and a chapter on Patterns Mining in the *CRC Handbook of Object Technology* published in January 1999. She is also the editor of a book on patterns experiences within her company, *The Patterns Handbook: Techniques, Strategies, and Applications*, published in 1998 by Cambridge University Press.