WILEY

TIMELY. PRACTICAL. RELIABLE.

Blueprints for High Availability

Second Edition

Evan Marcus
Hal Stern

Blueprints for High Availability

Second Edition

Evan Marcus
Hal Stern

WILEY
Wiley Publishing, Inc.

Blueprints for High Availability

Second Edition

Executive Publisher: Robert Ipsen
Executive Editor: Carol Long
Development Editor: Scott Amerman
Editorial Manager: Kathryn A. Malm
Production Editor: Vincent Kunkemueller
Text Design & Composition: Wiley Composition Services

Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data is available from the publisher.

ISBN: 0-471-43026-9

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

For Carol, Hannah, Madeline, and Jonathan
—Evan Marcus

For Toby, Elana, and Benjamin
—Hal Stern

Contents

Preface For the Second Edition

The strong positive response to the first edition of *Blueprints for High Availability* was extremely gratifying. It was very encouraging to see that our message about high availability could find a receptive audience. We received a lot of great feedback about our writing style that mentioned how we were able to explain technical issues without getting too technical in our writing.

Although the comments that reached us were almost entirely positive, this book is our child, and we know where the flaws in the first edition were. In this second edition, we have filled some areas out that we felt were a little flat the first time around, and we have paid more attention to the arrangement of the chapters this time.

Without question, our "Tales from the Field" received the most praise from our readers. We heard from people who said that they sat down and just skimmed through the book looking for the Tales. That, too, is very gratifying. We had a lot of fun collecting them, and telling the stories in such a positive way. We have added a bunch of new ones in this edition. Skim away!

Our mutual thanks go out to the editorial team at John Wiley & Sons. Once again, the push to complete the book came from Carol Long, who would not let us get away with slipped deadlines, or anything else that we tried to pull. We had no choice but to deliver a book that we hope is as well received as the first edition. She would accept nothing less. Scott Amerman was a new addition to the team this time out. His kind words of encouragement balanced with his strong insistence that we hit our delivery dates were a potent combination.

From Evan Marcus

It's been nearly four years since Hal and I completed our work on the first edition of *Blueprints for High Availability*, and in that time, a great many things

have changed. The biggest personal change for me is that my family has had a new addition. At this writing, my son Jonathan is almost three years old. A more general change over the last 4 years is that computers have become much less expensive and much more pervasive. They have also become much easier to use. Jonathan often sits down in front of one of our computers, turns it on, logs in, puts in a CD-ROM, and begins to play games, all by himself. He can also click his way around Web sites like `www.pbskids.org`. I find it quite remarkable that a three-year-old who cannot quite dress himself is so comfortable in front of a computer.

The biggest societal change that has taken place in the last 4 years (and, in fact, in much longer than the last 4 years) occurred on September 11, 2001, with the terrorist attacks on New York and Washington, DC. I am a lifelong resident of the New York City suburbs, in northern New Jersey, where the loss of our friends, neighbors, and safety is keenly felt by everyone. But for the purposes of this book, I will confine the discussion to how computer technology and high availability were affected.

In the first edition, we devoted a single chapter to the subject of disaster recovery, and in it we barely addressed many of the most important issues. In this, the second edition, we have totally rewritten the chapter on disaster recovery (Chapter 20, "A Disaster Recovery Plan"), based in part on many of the lessons that we learned and heard about in the wake of September 11. We have also added a chapter (Chapter 21, "A Resilient Enterprise") that tells the most remarkable story of the New York Board of Trade, and how they were able to recover their operations on September 11 and were ready to resume trading less than 12 hours after the attacks. When you read the New York Board of Trade's story, you may notice that we did not discuss the technology that they used to make their recovery. That was a conscious decision that we made because we felt that it was not the technology that mattered most, but rather the efforts of the people that allowed the organization to not just survive, but to thrive.

Chapter 21 has actually appeared in almost exactly the same form in another book. In between editions of *Blueprints*, I was co-editor and contributor to an internal VERITAS book called *The Resilient Enterprise*, and I originally wrote this chapter for that book. I extend my gratitude to Richard Barker, Paul Massiglia, and each of the other authors of that book, who gave me their permission to reuse the chapter here.

But some people never truly learn the lessons. Immediately after September 11, a lot of noise was made about how corporations needed to make themselves more resilient, should another attack occur. There was a great deal of discussion about how these firms would do a better job of distributing their data to multiple locations, and making sure that there were no single points of failure. Because of the economy, which suffered greatly as a result of the attacks, no money was budgeted for protective measures right away, and as

time wore on, other priorities came along and the money that should have gone to replicating data and sending backups off-site was spent other ways. Many of the organizations that needed to protect themselves have done little or nothing in the time since September 11, and that is a shame. If there is another attack, it will be a great deal more than a shame.

Of course, technology has changed in the last 4 years. We felt we needed to add a chapter about some new and popular technology related to the field of availability. Chapter 8 is an overview of SANs, NAS, and storage virtualization. We also added Chapter 22, which is a look at some emerging technologies.

Despite all of the changes in society, technology, and families, the basic principles of high availability that we discussed in the first edition have not changed. The mission statement that drove the first book still holds: "You cannot achieve high availability by simply installing clustering software and walking away." The technologies that systems need to achieve high availability are not automatically included by system and operating system vendors. It's still difficult, complex, and costly.

We have tried to take a more practical view of the costs and benefits of high availability in this edition, making our Availability Index model much more detailed and prominent. The technology chapters have been arranged in an order that maps to their positions on the Index; earlier chapters discuss more basic and less expensive examples of availability technology like backups and disk mirroring, while later chapters discuss more complex and expensive technologies that can deliver the highest levels of availability, such as replication and disaster recovery.

As much as things have changed since the first edition, one note that we included in that Preface deserves repeating here: Some readers may begrudge the lack of simple, universal answers in this book. There are two reasons for this. One is that the issues that arise at each site, and for each computer system, are different. It is unreasonable to expect that what works for a 10,000-employee global financial institution will also work for a 10-person law office. We offer the choices and allow the reader to determine which one will work best in his or her environment. The other reason is that after 15 years of working on, with, and occasionally for computers, I have learned that the most correct answer to most computing problems is a rather unfortunate, "It depends."

Writing a book such as this one is a huge task, and it is impossible to do it alone. I have been very fortunate to have had the help and support of a huge cast of terrific people. Once again, my eternal love and gratitude go to my wonderful wife Carol, who puts up with all of my ridiculous interests and hobbies (like writing books), our beautiful daughters Hannah and Madeline, and our delightful son Jonathan. Without them and their love and support, this book would simply not have been possible. Thanks, too, for your love and support to my parents, Roberta and David Marcus, and my in-laws, Gladys and Herb Laden, who *still* haven't given me that recipe.

Thanks go out to many friends and colleagues at VERITAS who helped me out in various ways, both big and small, including Jason Bloomstein, Steven Cohen, John Colgrove, Roger Cummings, Roger Davis, Oleg Kiselev, Graham Moore, Roger Reich, Jim "El Jefe" Senicka, and Marty Ward. Thanks, too, to all of my friends and colleagues in the VERITAS offices in both New York City and Woodbridge, New Jersey, who have been incredibly supportive of my various projects over the last few years, with special thanks to Joseph Hand, Vito Vultaggio, Victor DeBellis, Rich Faille, my roomie Lowell Shulman, and our rookie of the year, Phil Carty.

I must also thank the people whom I have worked for at VERITAS as I wrote my portion of the book: Richard Barker, Mark Bregman, Fred van den Bosch, Hans van Rietschote, and Paul Borrill for their help, support, and especially for all of those Fridays. My colleagues in the Cross Products Operations Groups at VERITAS have been a huge help, as well as good friends, especially Dr. Guy Bunker, Chris Chandler, Paul Massiglia, and Paula Skoe.

More thank-yous go out to so many others who I have worked and written with over the last few years, including Greg Schulz, Greg Schuweiler, Mindy Anderson, Evan Marks, and Chuck Yerkes.

Special thanks go, once again, to Pat Gambaro and Steve Bass at the New York Board of Trade, for their incredible generosity and assistance as I put their story together, and for letting me go back to them again and again for revisions and additional information. They have been absolutely wonderful to me, and the pride that they have in their accomplishments is most justified. Plus, they know some great restaurants in Queens.

Mark Fitzpatrick has been a wonderful friend and supporter for many years. It was Mark who helped bring me into VERITAS back in 1996, after reading an article I wrote on high availability, and who served as my primary technical reviewer and personal batting coach for this second edition. Thank you so much, Marky-Mark.

Last, but certainly not least, I must recognize my coauthor. Hal has been a colleague and a good friend ever since our paths happened to cross at Sun too many years ago. I said it in the first edition, and it's truer now than ever: This book would still just be an idea without Hal; he helped me turn just-another-one-of-my-blue-sky-ideas-that'll-never-happen into a real book, and for that he has my eternal respect and gratitude.

From Hal Stern

If Internet time is really measured in something akin to dog-years, then the 4 years since the first edition of this book represent half a technical lifetime. We've seen the rise and fall of the .com companies, and the emergence of networking as part of our social fabric, whether it's our kids sending instant

messages to each other or sipping a high-end coffee while reading your email via a wireless network. We no longer mete out punishments based on the telephone; in our house, we ground the kids electronically, by turning off their DHCP service. Our kids simply expect this stuff to work; it's up to those of us in the field to make sure we meet everyone's expectations for the reliability of the new social glue.

As networking has permeated every nook and cranny of information technology, the complexity of making networked systems reliable has increased as well. In the second edition of the book, we try to disassemble some of that complexity, attacking the problem in logical layers. While many of the much-heralded .com companies didn't make it past their first hurrahs, several now stand as examples of true real-time, "always on" enterprises: ebay.com, amazon.com, travel sites such as orbitz.com, and the real-time sportscasting sites such as mlb.com, the online home of Major League Baseball. What I've learned in the past 4 years is that there's always a human being on the other end of the network connection. That person lives in real time, in the real world, and has little patience for hourglass cursors, convoluted error messages, or inconsistent behavior. The challenges of making a system highly available go beyond the basics of preventing downtime; we need to think about preventing variation in the user's experience.

Some new thank-yous are in order. Through the past 4 years, my wonderful wife Toby, my daughter Elana and son Benjamin have supported me while tolerating bad moods and general crankiness that come with the author's territory. Between editions, I moved into Sun's software alliance with AOL-Netscape, and worked with an exceptional group of people who were charged with making the upper levels of the software stack more reliable. Daryl Huff, "Big Hal" Jespersen, Sreeram Duvvuru, and Matt Stevens all put in many hours explaining state replication schemes and web server reliability. Rick Lytel, Kenny Gross, Larry Votta, and David Trindade in Sun's CTO office added to my knowledge of the math and science underpinning reliability engineering. David is one of those amazing, few people who can make applied mathematics interesting in the real world. Larry and Kenny are pioneering new ways to think about software reliability; Larry is mixing old-school telecommunications thinking with web services and proving, again, that strong basic design principles stand up over time.

While on the software side of the house, I had the pleasure of working with both Major League Baseball and the National Hockey League on their web properties. Joe Choti, CTO of MLB Advanced Media, has an understanding of scaling issues that comes from hearing the (electronic) voices of millions of baseball fans. Peter DelGiacco, Group VP of IT at the NHL, also lives in a hard real-time world, and his observations on media, content, and correctness have been much appreciated. On a sad note, George Spehar, mentor and inspiration for many of my contributions to the first edition, lost his fight with cancer and is sorely missed.

Finally, Evan Marcus has stuck with me, electronically and personally, for the better part of a decade. Cranking out the second edition of this book has only been possible through Evan's herculean effort to organize, re-organize, and revise, and his tireless passion for this material. Scott Russell, Canadian TV personality, has said that if you "tell me a fact, I forget it; tell me the truth and I learn something; tell me a story and I remember." Thank you, Evan, for taking the technical truths and weaving them into a compelling technical story.

Preface from the First Edition

Technical books run the gamut from code listings sprinkled with smart commentary to dry, theoretical tomes on the wonders of some obscure protocol. When we decided to write this book, we were challenged to somehow convey nearly 15 years of combined experience. What we've produced has little code in it; it's not a programmer's manual or a low-level how-to book. Availability, and the higher concepts of resiliency and predictability, demand that you approach them with discipline and process. This book represents our combined best efforts at prescriptions for developing the disciplines, defining and refining the processes, and deploying systems with confidence. At the end of the day, if a system you've designed to be highly available suffers an outage, it's your reputation and your engineering skills that are implicated. Our goal is to supplement your skills with real-world, practical advice. When you see "Tales from the Field" in the text, you're reading our (only slightly lionized) recounts of experiences that stand out as examples of truly bad or truly good design.

We have sought to provide balance in our treatment of this material. Engineering always involves trade-offs between cost and functionality, between time to market and features, and between optimization for speed and designing for safety. We treat availability as an end-to-end network computing problem—one in which availability is just as important as performance. As you read through this book, whether sequentially by chapter or randomly based on particular interests and issues, bear in mind that you choose the trade-offs. Cost, complexity, and level of availability are all knobs that you can turn; our job is to offer you guidance in deciding just how far each should be turned for any particular application and environment.

We would like to thank the entire editorial team at John Wiley & Sons. Carol Long believed in our idea enough to turn it into a proposal, and then she coached, cajoled, and even tempted us with nice lunches to elevate our efforts into what you're reading now. Special thanks also to Christina Berry and Micheline Frederick for their editorial and production work and suggestions that improved the overall readability and flow of the book. You have been a first-rate team, and we owe you a debt of gratitude for standing by us for the past 18 months.

From Evan Marcus

This book is the product of more than 2 years of preparation and writing, more than 7 years of working with highly available systems (and systems that people thought were highly available), and more than 15 years of general experience with computer systems. Having worked in technical roles for consulting companies specializing in high availability and for software vendors with HA products, I found myself answering the same kinds of questions over and over. The questions inevitably are about getting the highest possible degree of availability from critical systems. The systems and the applications that run on them may change, but the questions about availability really don't. I kept looking for a book on this subject, but never could find one.

In 1992, I became intimately involved with Fusion Systems' cleverly named High Availability for Sun product, believed to be the very first high-availability or failover software product that ever ran on Sun Microsystems workstations. It allowed a predesignated standby computer to quickly and automatically step in and take over the work being performed by another computer that had failed. Having done several years of general system administrative consulting, I found the concept of high availability to be a fascinating one. Here was a product, a tool actually, that took what good system administrators did and elevated it to the next level. Good SAs worked hard to make sure that their systems stayed up and delivered the services they were deployed to deliver, and they took pride in their accomplishments. But despite their best efforts, systems still crashed, and data was still lost. This product allowed for a level of availability that had previously been unattainable.

High Availability for Sun was a tool. Like any tool, it could be used well or poorly, depending on the knowledge and experience of the person wielding the tool. We implemented several failover pairs that worked very well. We also implemented some that worked very poorly. The successful implementations were on systems run by experienced and thoughtful SAs who understood the goals of this software, and who realized that it was only a tool and not a panacea. The poorly implemented ones were as a result of customers not mirroring their disks, or plugging both systems into the same power strip, or running poor-quality applications, who expected High Availability for Sun to solve all of their system problems automatically.

The people who successfully implemented High Availability for Sun understood that this tool could not run their systems for them. They understood that a tremendous amount of administrative discipline was still required to ensure that their systems ran the way they wanted them to. They understood that High Availability for Sun was just one piece of the puzzle.

Today, even though the product once called High Availability for Sun has changed names, companies, and code bases at least three times, there are still people who realistically understand what failover management software (FMS) can and cannot do for them, and others who think it is the be-all and

end-all for all of their system issues. There are also many less-experienced system administrators in the world today, who may not be familiar with all the issues related to rolling out critical systems. And there are managers and budget approvers who think that achieving highly available systems is free and requires little or no additional work. Nothing so valuable is ever that simple.

The ability to make systems highly available, even without failover software, is a skill that touches on every aspect of system administration. Understanding how to implement HA systems well will make you a better overall system administrator, and make you worth more to your employer, even if you never actually have the chance to roll out a single failover configuration.

In this book we hope to point out the things that we have learned in implementing hundreds of critical systems in highly available configurations. Realistically, it is unlikely that we have hit on every single point that readers will run into while implementing critical systems. We do believe, however, that our general advice will be applicable to many specific situations.

Some readers may begrudge the lack of simple, universal answers in this book. There are two reasons for this. One is that the issues that arise at each site, and for each computer system, are different. It is unreasonable to expect that what works for a 10,000-employee global financial institution will also work for a 10-person law office. We offer the choices and allow the reader to determine which one will work best in his or her environment. The other reason is that after 15 years of working on, with, and occasionally for computers, I have learned that the most correct answer to most computing problems is a rather unfortunate, "It depends."

We have made the assumption that our readers possess varying technical abilities. With rare exceptions, the material in the book is not extremely technical. I am not a bits-and-bytes kind of guy (although Hal is), and so I have tried to write the book for other people who are more like me. The sections on writing code are a little more bits-and-bytes-oriented, but they are the exception rather than the rule.

* * *

When I describe this project to friends and colleagues, their first question is usually whether it's a Unix book or an NT book. The honest answer is both. Clearly, both Hal and I have a lot of Unix (especially Solaris) experience. But the tips in the book are not generally OS-specific. They are very general, and many of them also apply to disciplines outside of computing. The idea of having a backup unit that takes over for a failed unit is commonplace in aviation, skydiving (that pesky backup parachute), and other areas where a failure can be fatal, nearly fatal, or merely dangerous. After all, you wouldn't begin a long trip in your car without a spare tire in the trunk, would you? Busy intersections almost never have just one traffic light; what happens when the bulbs start to fail? Although many of our examples are Sun- and Solaris-specific, we have included examples in NT and other Unix operating systems wherever possible.

Throughout the book, we offer specific examples of vendors whose products are appropriate to the discussion. We are not endorsing the vendors—we're just providing their names as examples.

* * *

First and foremost, my gratitude goes to my family. Without the love, support, and understanding (or at least tolerance!) of my wife Carol and my daughters Hannah and Madeline, there's no way I could have written this book. A special note of thanks, too, to our family and friends, who pretended that they understood when I missed important events to stay home and write. See, it really was a book!

Additional thanks to Michael Kanaval (we miss you, Mike) for his inspiration and some excellent examples; to Joseph J. Hand, who helped with some of the NT material; to Michael Zona and John Costa for some of the backup stuff; to Mark Fitzpatrick and Bob Zarrow for some of my early and ongoing education in failover and general HA stuff; and to Mindy Anderson and Eric Burgener for clustering and SANs. Thanks, too—for general support, enthusiasm, and tolerance—to my parents Roberta and David Marcus, my in-laws Gladys and Herb Laden (now can I have that recipe?), and Ed Applebaum, Ann Sheridan, and Dinese Christopher, and everyone else at VERITAS and elsewhere who made suggestions and showed general enthusiasm and interest in this project. Special thanks to Mark Fannon and Evan Marks for excellent technical review and general help.

Thanks go out to the countless customers, users, and technical colleagues I've worked with over the years, with special thanks to the people at Morgan Stanley Dean Witter, Bear Stearns, Deutsche Bank, J. P. Morgan, Sun Microsystems, VERITAS Software, Open Vision, and Fusion Systems.

And a really big thanks to Hal Stern for being my personal door opener. In mid-1997 I finally made the decision to write this book. Having never written a book before, I knew that I needed help. I emailed Hal, looking for initial guidance from someone who had written a successful book. He wrote back and asked if perhaps we could collaborate. I thought long and hard for about 2 nanoseconds and then replied with an enthusiastic "Yes!" It was Hal's idea that we begin the writing process by creating a slide presentation. Our original set of 250 slides quickly grew to over 400, which we still present at technical conferences each year. By presenting the slides, we were able to determine what content was missing, where questions came up, and how the content flowed. It was a relatively (very relatively) easy job to then turn those slides into the book you see before you. Hal also originally contacted Carol Long at Wiley, got us on the agenda at our first technical conferences. This book would still just be an idea in my head without Hal.

From Hal Stern

My introduction to reliable systems began nearly 10 years ago, when I worked with the Foxboro Company to port their real-time, industrial control system from proprietary hardware to the Sun platform. You never really consider the impact of a hung device driver or failed disk drive until the device driver is holding a valve open on a huge paint mixing drum, or the disk drive is located along the Alaskan oil pipeline under several feet of snow. As the Internet has exploded in popularity, reliability and "uptime engineering" are becoming staples of our diet, because web surfers have caused us to treat most problems as real-time systems. As system administrators we have to decide just how much money to pour into reliability engineering, striving for four-nines (99.99 percent) or five-nines (99.999 percent) uptime while management remarks on how cheap hardware has become. There are no right answers; everything is a delicate balance of management, operations, money, politics, trust, and time. It's up to you to choose the number of nines you can live with. I hope that we help you make an informed choice.

This book would not have been possible without the love and support of my family. To my wife Toby and my children Elana and Benjamin, a huge thank-you, a big hug, and yes, Daddy will come out of his study now. I also want to thank the following current and former Sun Microsystems employees for educating me on various facets of availability and for their ideas and encouragement: Carol Wilhelmy, Jon Simms, Chris Drake, Larry McVoy, Brent Callaghan, Ed Graham, Jim Mauro, Enis Konuk, Peter Marcotte, Gayle Belli, Scott Oaks, and Wendy Talmont. Pete Lega survived several marathon sessions on complexity, recovery, and automation, and his inputs are valued. Chris Kordish and Bob Sokol, both of Sun Microsystems, reviewed the manuscript and offered their comments and guidance. Larry Bernstein, retired vice president of network operations at AT&T, challenged me to learn more about "carrier grade" engineering; it was an honor to have had discussions with a true Telephone Pioneer. Avi Nash and Randy Rohrbach at the Foxboro Company gave me a firsthand education in fault tolerance. Various individuals at Strike Technologies, Bear Stearns, Fidelity Investments, Deutsche Bank, Morgan Stanley Dean Witter, and State Street Bank proved that the ideas contained in this book really work. I thank you sincerely for sharing engineering opportunities with me, even if confidentiality agreements prevent me from listing you by name. A special thank-you to George Spehar, a true gentleman in every sense, for offering his sage management and economic decision-making advice. Ed Braginsky, vice president of advanced technology at BEA Systems, has been a good friend for eight years and a superb engineer for longer than that. His explanations of queuing systems, transaction processing, and asynchronous design, along with the thoughts of BEA cofounder Alfred Chuang, have been invaluable to me. Of course, thanks to Mom and Pop for teaching me the importance of being reliable.

Finally, a huge thank-you to Evan Marcus. We became acquainted while working on a customer project that required sniffing out performance problems during the wee hours of the morning. I'd never met Evan before, yet he was driving me around New Jersey and providing a steady patter at all hours. I should have recognized then that he had the stamina for a book and the power of persuasion to have me join him in the endeavor. Evan, thanks for your patience, understanding, and unique ability to prompt me out of writer's block, winter doldrums, and extreme exhaustion. It's been a pleasure traveling, working, and teaching with you.

About the Authors

Evan Marcus is a Principal Engineer and the Data Availability Maven at VERITAS Software. His involvement in high-availability system design began in 1992 when he codesigned a key piece of the first commercial Sun-based software for clustering. After a stint as a system administrator for the equities trading floor at a major Wall Street financial institution, Evan spent over 4 years as a sales engineer at VERITAS Software, servicing all sorts of customers, including Wall Street financial firms. Since then he has worked in corporate engineering for VERITAS, consulting and writing on many different issues including high availability, clustering, and disaster recovery. He has written articles for many magazines and web sites, including, most recently, TechTarget.com, and is a very well-regarded speaker who contributes to many industry events. Since completing the first edition of *Blueprints*, he was an editor and contributing author for *The Resilient Enterprise,* a 2002 VERITAS Publishing book on disaster recovery that was the first VERITAS published book that involved a collaboration of industry authors. Evan holds a B.S. in Computer Science from Lehigh University and an M.B.A. from Rutgers University.

Hal Stern is a Vice President and Distinguished Engineer at Sun Microsystems. He is the Chief Technology Officer for Sun Services, working on design patterns for highly reliable systems and networked applications deployed on those systems. In more than 10 years with Sun, Hal has been the Chief Technology Officer for the Sun ONE (iPlanet) infrastructure products division and the Chief Technologist of Sun's Northeast U.S. Sales Area. Hal has done architecture, performance, and reliability work for major financial institutions and electronic clearing networks, two major professional sports leagues, and several of the largest telecommunications equipment and service companies.

Hal served as contributing editor for *SunWorld Magazine* for 5 years, and was on the editorial staff and advisory board of IDG's *JavaWorld* magazine. Before joining Sun, Hal developed molecular modeling software for a Boston area startup company and was on the research staff at Princeton University. He holds a B.S. in Engineering degree from Princeton University. When not at the keyboard, Hal coaches Little League, plays ice hockey, cheers for the New Jersey Devils, and tries desperately to golf his weight.