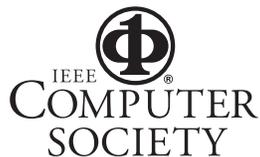


Understanding the Power of Abstraction in Patterns

Linda Rising

Vol. 24, No. 4
July/August 2007

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.



© 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

For more information, please see www.ieee.org/web/publications/rights/index.html.

Understanding the Power of Abstraction in Patterns

Linda Rising, *independent consultant*

An unusual experience report on mining and documenting patterns reveals the value of higher-level abstraction and of lower-level implementation in a different domain.

We are all, pattern authors or not, limited by our own experience. We tend to see the solutions we've applied as the most useful and, therefore, the ones we want to document and share as patterns. Rarely do we take the time to look around, either at a higher level of abstraction or a related domain. We just don't have the time. Pattern writing consumes a lot of energy and, for those in the patterns community, the work involves shepherding (reviewing) conference papers and attending conference workshops.

Such shepherds and workshop participants offer another pair of eyes and another set of experiences when verifying solutions in fledgling patterns. So, when Mary Lynn Manns and I were working on a collection of patterns for our book,¹ we submitted subsets of our growing collection to conferences for review. Writers' workshops at the conferences provided good feedback on the patterns and connected us to other change agents who had tried to introduce patterns in their teams and organizations. These connections were invaluable as the pattern collection grew, providing both a sounding board for our ideas and a different viewpoint on the patterns and collection.

What the conference reviewers couldn't provide was a detailed examination of a related domain, simply because it takes too much time and energy for what is typically seen as a low return on investment. However, at one point, various reviewers began to say the patterns we

were writing were too low-level. We were writing about introducing patterns, but the reviewers felt that our patterns could be used to introduce any new idea. Our exploration of this suggestion took us on a journey outside the software domain that gave us a broader and deeper understanding of our patterns.

Resisting abstraction

Determining the appropriate level of abstraction is an old debate in the patterns community—authors are always asking, "Where should abstraction end?" I recall a discussion in one company's writers' workshop in which someone in one functional area had documented a pattern for a particular application even though the pattern had already been documented elsewhere for a different application. The question was, Should we write one larger, more abstract pattern that captures the ideas in both lower-level patterns? Or, should we write, and add to the pattern library, two nearly iden-

tical patterns that really solve the same problem but have different known uses, examples, and application areas?

I felt (echoing Einstein) that the answer was to make patterns as low-level as possible but no lower. Readers, it still seems to me, respond to the vocabulary of their own domains. I think the clearest example of this is the motivation, known uses, and programming languages used in the Gang-of-Four book.² The developers I worked with were initially enthusiastic about the book's patterns but were surprised they didn't clearly address performance issues. This was important for other users as well—in particular, for the Ada programming language community, a group that the patterns community has (even now) missed completely.

Ada programmers looked at the GoF book and said, “C++? Smalltalk? Windows stuff? This can't be for me!” and moved on. These developers weren't going to spend time trying to translate abstractions from a book without Ada or real-time, embedded examples, so they didn't see any use for patterns. A few Ada folks made forays into the patterns world but were never able to bridge the gap.

What clinched it for me was reading Joseph Bergin's pattern Do the Right Thing (see the sidebar). This pattern pokes fun at the idea of high-level abstraction by suggesting that if we just did the right thing, that's the only pattern we would need! All other patterns are just implementations of this one grand problem and solution.

As a result of this early bias, Manns and I resisted the idea of making our patterns more abstract. We resisted not only on principle but also because we felt we'd be demonstrating a certain lack of integrity, since our experience as successful change agents was based (we believed) only on introducing patterns. We didn't have any experience with introducing other new ideas, and we hadn't solicited such stories from others.

Patterns for introducing new ideas

During this time of reflection regarding the abstraction issue, my daughter, Amy, was the programs director for the Multiple Sclerosis Society in Nashville. For Christmas, she gave me the biography of Sylvia Lawry, the founder of the MS Society.³ I'm always curious about people, especially women, who get things going, so I thought this would be an interesting read. I was totally unprepared,

Joseph Bergin's “Do the Right Thing”

Context:

Things are bad. Really bad.

Forces:

When things are bad it is really tough and bad things happen.

When things get better the bad stuff doesn't happen any more and you feel good. Really good.

Solution:

Do the right thing. Make the bad thing better.

Resulting Context:

Things are good. Really good.

Known Uses:

When you were small your father would make the Monsters Under the Bed go away just by sticking his head in your room. He did the right thing.

When you are really sick, eat your Mom's chicken soup. Only your Mom's. Only she knows how to do the right thing.

Related Patterns:

Deep Thought, Law of Unintended Consequences

With thanks to Spike Lee, who, as Mookie in his film Do the Right Thing, did the right thing. Decisively.

With thanks to my sponsors, the developers of Prozac, for a generous grant (product), which made this possible.

Note and disclaimer. If anything can be a pattern, then this is. If patterns are just a literary form, then this is. However, if patterns are about resolution of forces and solving problems in context, then this is just fun. And no, I did not actually have a grant from Eli Lilly and Company. That too was just fun. Enjoy.

however, to see throughout the book examples of the patterns that Manns and I had been writing. In many cases, small changes in wording made the text closely match the stories we had collected from change agents in software teams who had tried to get their peers to use the Mediator pattern. The book didn't always provide enough detail for us to use its stories and known uses, but we extracted enough information to convince us that the reviewer feedback we had received was correct. We began to think of our patterns as appropriate for introducing any new idea, not just patterns.

I don't believe the debate over abstraction levels is over—there's no simple answer.

And then new stories began to appear—stories about introducing agile practices, new testing approaches, CMM levels, CM practices ... you name it. And they fit. Expanding all the patterns in the collection to this higher level almost always required little more than changing a few words and adding new known uses.

It was the “right thing to do,” as Bergin would say. It felt right, and we weren't losing anything by moving our focus to a higher level of abstraction. We were able to include all the information about other new ideas without significant changes to the existing patterns. In fact, the new stories added credibility.

We also found that we had experience we weren't aware of. Manns is an academic at a university going through a lot of change, so she shared her experiences as a successful change agent introducing innovation into a changing organization. We even convinced one of her colleagues to write a case study about introducing a particular innovation in his college. I began introducing Scrum practices in my organization. As we looked around, we saw that we were using our patterns in many other instances, even in our personal lives. How many times had we tried to convince our friends and family to try something new?

So, we expanded our focus and wrote *Fearless Change: Patterns for Introducing New Ideas*. Were there downsides to the increased abstraction? Of course—as with the solution to any pattern, there were consequences. Readers looking for solutions to particular needs would see our stories or known uses, which were outside their domain, and say, “But this is about <fill in the blank>, while I'm interested in <fill in more blanks>.” We're all stuck in our own domains. However, most readers have been able to apply our patterns without stumbling over a particular story or known use, so the consequences have been minor.

Manns and I have also been able to adapt our patterns as needed. For example, in our training classes on the *Fearless Change* patterns, participants act out a play. We provide scripts where the actors tell the story of a change agent who uses the patterns to introduce agile practices to an organization. When the situation calls for it, we can easily adapt the play. I gave a presentation based on the play where, instead of agile practices, Play-Doh modeling was the innovation. I have also written a version for pharmaceutical represen-

tatives, because my daughter now works for a pharmaceutical company that uses *Fearless Change* for training purposes. Introducing agile practices isn't that different from a rep trying to influence a doctor to write a prescription for a particular drug.

Despite these experiences, I don't believe the debate over abstraction levels is over—there's no simple answer. Continuing to raise abstraction leads to Bergin's single pattern, so we still face the question of when to stop. And, as I describe next, looking at the lower levels is still valuable.

Looking to another domain

Recently, I visited a college's campus bookstore. (I always enjoy stopping in college bookstores, not only to experience sticker shock over the price of textbooks but also to see what courses are being offered these days. It always makes me want to go back to school—and also glad I don't have to!) As I headed down one aisle, a large copy of *Diffusion of Innovation* by E.M. Rogers⁴ caught my eye—“strange,” I thought, because I was in the agriculture section. We heavily referenced this publication in our book because Rogers introduces a model that segments a “normal” population, based on its response to innovation, into the following categories: Innovators, Early Adopters, Early Majority, Late Majority, and Laggards.

I was intrigued by the accompanying text for the course, a small paperback called *Two Ears of Corn* by Roland Bunch.⁵ The title is from Jonathan Swift's *Gulliver's Travels*:

Whoever could make two ears of corn, or two blades of grass, to grow upon a spot of ground where only one grew before, would deserve better of mankind, and do more essential service to his country than the whole race of politicians put together.

As I leafed through the book, which was subtitled “A Guide to People-Centered Agricultural Improvement,” I was surprised to see that it described the patterns in *Fearless Change*. This wasn't a book about patterns, however. It was for a course titled, “The Diffusion and Adoption of Agricultural Innovations.” The course description read: “Factors that influence rates of diffusion and adoption of innovations. Consequences of adopting or rejecting innovations. Processes by which

change agents influence introduction and adoption of innovations.”

So I bought the book, which turned out to be pretty easy to read, but, as a “pattern junkie,” I kept tripping over contexts, forces, problems, and solutions. I couldn’t help but think that the book would have been even more useful and easier to understand and implement if it had been cast as a collection of patterns.

I was especially moved by the following:

The introduction of innovations into Third World agriculture has met with everything from disaster to exhilarating success. Well-bred animals have often died of disease and malnutrition. Home and school vegetable gardens have yielded disappointing results in many projects in India and nearly everywhere they have been tried in Latin America. Yet poor goatherds in a remote program area in the Bolivian Andes have walked for fourteen hours to buy animal vaccines, and Indian farmers involved in a program in Guatemala are producing, with their own native varieties, up to 3,200 kilos per hectare of dry beans, twice the average yield in the United States.

Some innovations increase the production of thousands of farmers while others fail to be accepted by even a handful. If we are going to work with only a few innovations, how can we choose the ones that will find the widest acceptance?

World Neighbors’ experience indicates that there are a number of widely applicable criteria that can guide us in choosing the appropriate technology for any particular area.

“Widely applicable criteria”—patterns! I was so excited by this book and the idea that the patterns in *Fearless Change* might help a large part of the population outside the original target audience. I also felt that this was the kind of “big idea” application that many of us were looking for—a way to help make the world a better place.

I contacted Roland Bunch and learned that *Two Ears of Corn* has been translated into eight languages, and more than 60 nongovernmental organizations, government agencies, and universities around the world use it. I asked if I could contribute by producing a patterns handbook for agricultural innovation,

with all royalties going to the nonprofit organization that sponsored the original publication. *Two Ears of Corn*, with its stories, simple drawings, and appealing message, helps third world farmers improve their lives. Who wouldn’t want to step up and contribute?

So, when I first started rewriting *Two Ears of Corn*, my goal was to give something to a cause. However, as I read and re-read the book and began to weave patterns from Bunch’s words, I found nuggets of wisdom that don’t translate to the patterns in *Fearless Change*. In other words, I’m receiving as much from my translating work as I had hoped to give. *Two Ears of Corn* has helped Manns and me as we continue to work on expanding and improving our patterns. When, for example, one of us stumbles across a new pattern that we’re not sure we captured, the clear and compelling stories of agricultural innovation in small villages enable us to think more clearly about context, forces, problem, and solution. I’m learning about the patterns Manns and I have written as I struggle to understand a domain I know nothing about. I’m not an agricultural expert, but I have learned how people react to change in a new setting, and the wisdom in *Two Ears of Corn* resonates with me.

Patterns and vocabulary

Pattern writers often think of their work as producing a language, where the patterns’ names have a connection with each other that we can structure as a network. The pattern names frame a vocabulary for the target domain, letting users converse about the domain in a code that simplifies the discussion.

I once asked Martin Fowler about two patterns that Manns and I were discussing. We couldn’t decide whether to collapse them into one pattern or to keep them separate. Fowler said that having extra patterns increases the language’s size and results in more effort for users to remember and apply the language. I thought this was an excellent answer. The patterns provide a vocabulary. If a particular pattern contributes to that vocabulary so that essential ideas would be missing from appropriate conversations, the pattern should be included. Not having certain words means you can’t have certain conversations. You can clearly see this at work in the following story.

In researching third world environments and related topics, I stumbled across a report



**Not every
discovery
will lead
to a pattern
worth adding
to our
collection.**

about the Pirahãs,⁶ a small tribe of approximately 300 people living on the Maici River in Brazil. This tribe has an unusually simple language—so simple that it has attracted the attention of numerous linguists, anthropologists, and cognitive researchers.

For example, the Pirahãs have no way of expressing numbers. Researchers have never heard words for “all,” “every,” or “more” from this tribe. There is one word, *bóí*, which comes close to the numeral 1, but it can also mean “small.” The tribal members don’t even appear to count without language—on their fingers, for example.

One researcher noted that not having words for numbers seems to mean that there is no thought about numbers.⁶ His findings have stirred up the old controversy originally put forth by linguist Benjamin Whorf, who said that people can only construct thoughts for which they possess words.⁶ Because the Pirahãs have no words for numbers, they can’t understand the concept of numbers.

By contrast, a similar tribe, the Warlpiri—a group of Australian aborigines whose language is a bit more advanced than the Pirahãs (they recognize “one,” “two,” and “many”)—learned to count beyond three in English. However, over eight months, another researcher tried in vain to teach the Pirahãs to count in Portuguese. “In the end,” he said, “not a single person could count to ten.”⁶

It’s not that the Pirahãs lack intelligence. “Their thinking isn’t any slower than the average college freshman,” says one researcher.⁶ And the Pirahãs don’t live in isolation. They mix with people from the surrounding populations, so their intellectual abilities are likely equal to those of their neighbors. They just don’t use numbers and don’t see any use for them.

Similarly, not having a particular pattern means that users can’t include that idea in a discussion of problems and solutions in that domain. As I learn about agriculture, I see missing patterns that Manns and I might want to add to our collection. The question we face isn’t whether the missing ideas represent patterns but whether they contribute useful vocabulary about our domain. Once we agree on that, capturing the pattern is easier.

We don’t often make decisions about the importance of ideas with the goal of discarding those things that are not useful. Most of us, especially now that we live in Internet

Land, accumulate everything we find—feeling somehow that we will find a use for it. Perhaps, like the Pirahãs, it might pay us to become more discriminating in our lives and in our pattern languages to produce cleaner, truly useful vocabularies instead of growing complicated networks of patterns that we document just because we can.

One example of a pattern from *Two Ears of Corn* that hasn’t made it into the second edition of *Fearless Change* is “Broader Human Goals”:

Most programs have much broader goals than simply increasing agricultural productivity. They aim to improve the quality of life, liberate the human spirit, lead more fulfilling lives, and develop the total human being. What this means is hard to define, and should not be defined by the program, but with the help of the villagers. The villagers may not want to face these issues until they have overcome their more immediate problems. Yet, if these more subtle goals remain undefined for too long, they may remain unfulfilled.

Our experience shows that even programs with no other goal beyond agricultural improvement should pay some attention to the broader human goals. Agricultural improvement becomes slow, inefficient, and temporary if broader human factors are not taken into account. If agricultural work makes people more individualistic, dishonest, and self-seeking (as it sometimes does), the people’s ability to work together, which permanent agricultural improvement requires, will be destroyed. If agriculture teaches villagers to feel incapable and to depend on outsiders, the process of agricultural improvement will end when the program does. The design of every aspect of the program—from leadership training and administrative style to the choice of the technology to be taught—must take into account the impact it will have on the achievement of these broader human goals.⁵

When we bring patterns or agile practices to our work communities, we might have in the recesses of our minds some idea about improving life for our team, but we typically don’t address “broader human goals”—except, perhaps, in a retrospective. I believe this is an important pattern for introducing agri-

cultural improvement in the third world. It might be that we're underestimating its importance in software development, but we have no plans to document it—yet. I would be interested in hearing from readers with strong opinions (one way or the other) about this. This is another ongoing discussion within the patterns community.

Patterns—their names, presence or absence, level of abstraction, and target domain—affect the way we think about our environment and how we solve problems in it. Having a pattern (or name) for something means we can think and talk about it. Sometimes the effort in facing problems is reduced by moving to a higher level of abstraction. It can also be beneficial to reach across domains so that, for example, expertise in third world agriculture can help us understand how to introduce new software development ideas.

To write the most effective pattern languages, we must be flexible and open enough to exploit the power of abstraction when it's appropriate and to learn from any domain's implementation details, even if the connection seems remote. At the same time, we must be wise enough to see that not every discovery will lead to a pattern worth adding to our collection. Seeing each new idea clearly and examining it from all sides to identify its useful application will help us learn more about the problems we face. Maybe this is what learning is all about, and maybe this is how we keep building. ☺

Acknowledgments

My story is one of many “happy accidents.” The patterns I helped write are better because of many

About the Authors



Linda Rising is an independent consultant who conducts presentations on patterns, retrospectives, agile processes, and change. Her background includes university teaching and industry work in telecommunications, avionics, and strategic weapons systems. She received her PhD in computer science from Arizona State University, focusing on object-based design metrics. She's a member of the IEEE Computer Society, ACM, Hillside Group, and Agile Alliance. Contact her at risingl@acm.org.

shepherds and countless workshop participants and because I was lucky enough to find a sympathetic and energetic coauthor—my good friend, Mary Lynn, who also provided suggestions for this article. I was also lucky enough to have fellow pattern enthusiasts such as Joe Bergin, who deserves recognition for writing *Do the Right Thing* and for providing feedback on this article. I also thank the three anonymous reviewers whose comments helped me make improvements. I dedicate this work to all these people.

References

1. M.L. Manns and L. Rising, *Fearless Change: Patterns for Introducing New Ideas*, Addison-Wesley, 2005.
2. E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Systems*, Addison-Wesley, 1995.
3. M. Dugan, *Courage: One Woman's Dream and the Mighty Effort to Conquer Multiple Sclerosis*, Ivan R. Dee, 2001.
4. E.M. Rogers, *Diffusion of Innovations*, 4th ed., The Free Press, 1995.
5. R. Bunch, *Two Ears of Corn*, World Neighbors, 1995.
6. P. Gordon, “Numerical Cognition without Words: Evidence from Amazonia,” *Science*, vol. 306, no. 5695, pp. 496–499; <http://faculty.tc.columbia.edu/upload/pg328/GordonSciencePub.pdf>.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

To receive regular updates, email

dsonline@computer.org



IEEE
distributed systems
Expert-authored articles and resources
ONLINE

VISIT THE IEEE'S
FIRST ONLINE-ONLY
DIGITAL PUBLICATION

IEEE Distributed Systems Online brings you peer-reviewed features, tutorials, and expert-moderated pages covering a growing spectrum of important topics including Grid Computing, Mobile and Wireless, Middleware, Distributed Agents, and more.

dsonline.computer.org