

Patterns: Spreading the Word

by
Linda Rising

Software engineers are always casting about, examining other engineering disciplines, looking for solutions to the now decades-old problem called the software crisis. They have considered hardware, industrial engineering, manufacturing scenarios - the list goes on.

The latest investigation into a perhaps-related area is an especially provocative one. A group of respected individuals in the object-oriented community have been taking a hard look at the work of Christopher Alexander, an architect, a building architect. Alexander has some intriguing observations about an entity he calls a pattern, defined as follows [1]:

"Each pattern describes a problem that occurs over and over again in our environment and then describes the core of the solution to that problem in such a way that you can use this solution a million times over without ever doing it the same way twice."

Software patterns have been defined by Jim Coplien and Doug Schmidt [2]:

"Design patterns capture the static and dynamic structures of solutions that occur repeatedly when producing applications in a particular context."

Since members of the software engineering community have been searching, unsuccessfully for the most part, for an answer to the problem of re-inventing the wheel, the patterns investigation has been attracting considerable attention.

I have been an active patterns advocate since I attended the OOPSLA '94 tutorial last October and purchased the GOF text (the Gang of Four: Gamma, Helm, Johnson, and Vlissides) [3]. This paper presents an account of my attempt to spread the word about patterns at my company, AG Communication Systems.

I started by giving noon-time brown bag presentations on object-oriented design heuristics, based on another tutorial from OOPSLA, and during these, mentioned the notion of design patterns to obtain some initial feedback.

Acting on a suggestion from my manager, I scheduled a presentation for our vice-president and his staff (the Gang of Sixteen, GOS) to get approval to buy copies of the GOF text and start training in the GOF patterns, with the long-term goal of collecting domain-specific patterns in a "Best Practices" handbook.

I repeated this presentation at our monthly OO Technology Forum to stimulate interest in the training. I also posted a report from the trenches to the patterns listserver [4]. Thanks to the overwhelming response from patterns lovers everywhere, I felt I was not alone in struggling to introduce patterns to my company. I received several requests for the introductory materials I had created, so I posted an ASCII version of my presentation.

The GOS requested an evaluation phase. Members of the evaluation team included designated representatives of the GOS, some of our resident OO gurus, and others who were just interested in patterns. There were about twenty team members in all.

We ordered copies of the GOF text for the team and others around AG Communication Systems. This part is key. Put the book in the hands of knowledgeable designers and patterns will begin to take hold. It's not enough to derive a lot of benefit, but it is a powerful initial step. My advice to correspondents on the listserver was: start spreading the word by ordering books. Get enough for an evaluation team and more because you'll find that others will want a copy. That's the biggest selling point for patterns - there is a book and designers will start reading it!

The evaluation phase comprised four one-hour meetings over a two-week period. We covered the five patterns and the example in Ralph Johnson's paper [5]. The evaluation meetings were open to anyone who was interested in patterns. That provided a larger audience for the information and more input for the discussions. The papers [6-17] listed in the references were also made available.

It was a thoroughly enjoyable experience - the chance to get together with other developers and talk about design problems and possible patterns solutions. This kind of self-examination doesn't often occur in today's pressured environment. Typically, we rush headlong into the next project without stopping to adequately reflect on what we should have learned from the last. As a result, we could make the same mistakes over and over.

Some members of the team expressed their concern with the lack of real-time, embedded system considerations in the GOF text. These are not trivial enhancements and I anticipate that someone will provide them in the future. Until then, we're building on the GUI examples in the motivation. It's true that everyone can understand this context and that's a good beginning.

The evaluation phase was easy. The patterns sell themselves. The team had no trouble answering the questions:

1. Is the notion of patterns a worthwhile technology for AG Communication Systems? The consensus was an overwhelming yes!
2. Are the patterns in the GOF text worth a training investment? Again, the consensus was an overwhelming yes!

I went back to the GOS with the results and the following suggestions:

1. Every interested designer should have a copy of the GOF text.
2. We should provide training for teams, to enhance communication and design knowledge.
3. I would continue the noon-time brown bag meetings, covering the rest of the patterns, to keep the momentum and provide help for teams who really needed this

information. The word was out and patterns were already appearing on some projects.

At this point, we were fortunate to find a training consulting organization that would come in for a trial with our evaluation team. The training was based on the GOF text and spanned an intensive three days. The evaluation team liked the course but found that twenty-three patterns in three days was too much. They also made some minor suggestions about format and the lab exercise(s).

Ralph Johnson has observed that [1] "... people can't learn patterns without trying them out. Also, people need to find them in their own problem domain." Ralph is absolutely right about the difficulty of learning patterns from lecture. The trial offering of the course uncovered that problem - even though all the members of that group had been through the GOF text in the brown bags.

Fortunately, the trainer spent time after class working with some of the students to apply patterns to their design problems. This experience was so valuable that we wanted to incorporate this consulting opportunity in the training package and reduce the concentrated patterns presentations. The class schedule was changed from three full days to one full day followed by four half-days of training, allowing for four half-days of consulting. We tried to schedule each course offering for a specific team to provide the biggest leap in abstraction and communication. In most cases, however, teams felt that having everyone out for training at the same time was not possible, so they split into two or more sessions. This meant the consulting time was open to all the teams. Some couldn't take advantage of it and did not benefit as much.

Some of the questions raised during this part of the adventure included: What should we do next? Do we start patterns mining and how is that done? We also have another side of the house that is not object-oriented. Managers of those developers were concerned that they were being left out of all the patterns activity. The GOF text was object-oriented; the training was object-oriented. How could they be a part of this new technology?

The answers to these questions came in the guise of an exceptional fellow at AT&T, Jim Coplien (Cope), and his colleague, Bob Hanmer, who introduced us to writers workshops. Cope showed us a new way to look at patterns. Cope and Bob are patterns mining the 4ESS, one of AT&T's large telecommunications switches. The patterns they shared with us had no object diagrams or C++ code. Furthermore, some of our developers recognized some of the patterns as part of the object-oriented projects and the GTD-100, the large switch we develop for GTE and independent telephone companies.

As a result of this experience, we have started patterns mining on the GTD-100 with the help of resident domain experts. We hope to share knowledge of patterns across the company and with the folks on the 4ESS.

Ralph Johnson [1] said, "You'll find that most people are not good at finding patterns. Fortunately, you don't need (or want) most people to be doing that. You want to make everybody feel included and wanted, but ultimately people will sort

themselves out into those who will want to work on finding patterns and those who don't. ... your organization's real job is to get a product finished, and patterns are a means to that end. Most people should just get on with their work."

We held writers workshops to look at some of the ESS patterns. Thus far, most of the patterns have also been seen on the GT -5 , so even if they weren't written here, they should be familiar to many of our developers. Some modification will be required for these patterns before they become ours.

Ralph Johnson comments on this activity [1]: "Working with people from AT T who are working on similar problems is of course an outstanding idea. That is probably where you should put your energy for finding new patterns, because there will be a lot of synergy. Moreover, the best way to find patterns is to compare and contrast similar systems. So, you should take advantage of this opportunity."

We have a local patterns home page, which is currently behind a firewall. The patterns from the ESS , from Cope's writers' workshop, and the patterns we have found on the GT -5 are referenced here each allows input from readers to enable the evolution and improvement of these patterns. This should allow us to create evolving, changing patterns that would eventually be part of a " Best Practices" handbook. There are also several links to other patterns locations [1].

We are also establishing a patterns mining process. We are encouraging developers to write their own patterns, attempting to answer the questions, "What knowledge would be lost to the company if I were to leave tomorrow? What do I know that I have done a thousand times that I think everyone already knows?" Patterns then go through a writers workshop [2]. After updates to the pattern have been made, the pattern is posted on our home page to evolve as others read the pattern and make comments. After a suitable interval, some of these patterns would become part of the " Best Practices" handbook and used for training newcomers to projects and the company.

We are adopting Cope and Job's approach of interviewing domain experts who don't have the time to write patterns. The interviews are taped and used for patterns creation by pattern crafters. After a pattern has been identified, it goes back to the domain expert, then through a writers workshop and on to the Web page.

I feel the time is right for patterns. People in the object-oriented community are looking for help and, while patterns are not the silver bullet, they offer some help with just a little investment.

There's an interesting side-effect in the patterns activities. Developers at AG Communication Systems feel they're really on the cutting edge and they are! It's exciting to have famous visitors who share what they have learned. There's a sudden interest in attending conferences and classes, including Ralph Johnson's course on patterns at the University of Illinois, this year's P oP, and OOPS A ' 5. When asked about productivity benefits from patterns, I have pointed out that people who are excited about their work and feel a part of something stimulating and rewarding will be more productive, regardless of the technology that's involved. I'm very happy to be a part of something that brings this kind of benefit to my fellow

workers.

References

1. Alexander, Christopher, et al. A Pattern Language, Oxford University Press, New York, 1977.
2. Coplien, James O. and Douglas C. Schmidt, ed. Pattern Languages of Program Design, Addison-Wesley, 1995.
3. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley, 1995.
4. <http://st-www.cs.uiuc.edu/users/patterns/Lists.html>
5. Johnson, Ralph E. "How Patterns Work in Teams," ROAD, September-October 1994.
6. Booch, Grady. "Patterns," Object Magazine, July-August 1993.
7. Coplien, James O. "Curiously Recurring Template Patterns," C Report, February 1995.
8. Coplien, James O. "Generative Pattern Languages: An Emerging Direction of Software Design," Proceedings of the 4th Annual Borland International Conference, June 1994.
9. Coplien, James O., "Setting the Stage," C Report, October 1994.
10. Coplien, James O. "Software Design Patterns: Common Questions and Answers," AT&T Bell Laboratories report.
11. Coplien, James O. "What I Did On My Summer Vacation," C Report, September-October 1994.
12. Huni, Hermann, Ralph Johnson, and Robert Engel. "A Framework for Network Protocol Software," submitted for OOPSLA '95.
13. Goenig, Andrew. "Patterns and Antipatterns," JOOP, March-April 1995.
14. Lea, Doug. "Christopher Alexander: An Introduction for Object-Oriented Designers," Software Engineering Notes, Vol. 19, No. 1, January 1994.
15. Schmidt, Douglas C. and Paul Stephenson. "Using design patterns to evolve system software from MFC to Windows NT," C Report, March-April 1995.
16. Vlissides, John. "Pattern hatching - Perspectives from the "Gang of Four," C Report, March-April 1995.

17. Vlissides, John, "Reverse Architecture," IBM T.J. Watson Research Center report.

18. Johnson, Ralph E., Electronic mail message.

"Patterns: Spreading the Word," Object Magazine, Dec. 1996, pp. 5-57.

© 1997 by SIGS Publications, Inc., used with permission.

© 2001 AG Communication Systems, used with permission.