The Language of Shepherding

A Pattern Language for Shepherds and Sheep

Neil B. Harrison
Lucent Technologies
11900 North Pecos st.
Denver, Colorado 80234
(303) 538-1541
nbharrison@lucent.com

Many of us have submitted patterns to our colleagues for feedback prior to a writers' workshop or other type of review. In fact, at PLoP conferences, all submissions are subjected to shepherding before they are evaluated for acceptance. Unfortunately, the quality of shepherding varies widely. Some people receive extremely helpful comments, but others receive only cursory remarks, and a "Looks good" endorsement.

Yet shepherding can be a very powerful tool for improving patterns. It can go well beyond hints for grammar and usage, even to the heart of the work being shepherded. In fact, shepherding can turn a paper about a solution into a pattern. But it requires more than a casual reading by the shepherd: it requires attention, and actions such as the ones described in the patterns below.

These patterns have been gleaned from significant experience with shepherding, both as a shepherd and as a "sheep". This experience is augmented with experience in giving and receiving feedback for non-pattern technical works, from teaching, and from observing. The patterns should help you become a better shepherd, both in a formal pattern shepherding setting and in other situations.

Setting the Stage

Imagine that you have agreed to be a shepherd for an upcoming PLoP. Obviously, you have had some experience with shepherding; at the very least, you have been a "sheep" yourself. You know what it feels like. In addition, you know about patterns. You have an idea of what makes up a good pattern. Ideally, you have been to writers' workshops, perhaps at a PLoP.

You just received mail from the program chair informing you of your shepherding assignment. She notes that you have not quite a month before the final papers are due, so you need to start soon. So you immediately take a look at the paper you have been assigned. As you read it, you recognize your responsibility – your help may make the difference between acceptance and rejection. And you want to do the best job for the author that you can. Now what?

It is likely that you will run across several of the problems described below. Of course, since you are up on the latest pattern literature, you have read these patterns, and are ready to apply them to help the author improve the work.

A Map of the Patterns

The patterns here can be grouped into two major categories that interact very closely with each other. The first group are patterns that deal with the shepherding process itself, the tactics of shepherding. These patterns are (numbers refer to the pattern numbers in the language):

1. *Three Iterations:* How to budget your time and effort to make shepherding effective.
2. *The Shepherd Knows the Sheep:* How to establish a productive relationship between you and the author.
3. *Half a Loaf:* How to make sure that shepherding continues to move forward.
4. *Big Picture:* How to grasp the gist of the pattern right off the bat.

5. *Author as Owner:* How to keep from writing the pattern for the author.
8. *Forces Define Problem:* How to understand the problem at a deeper level.

The shepherding process is essentially a reviewing process, as is the writers' workshop. Therefore, it should come as no surprise that these patterns are closely tied to "A Pattern Language for Writers' Workshops," [1] Selected patterns will be referenced with thumbnails, and will be noted as Workshop patterns.

Since shepherding is all about improving the pattern itself, the second group of patterns deals with aspects of the pattern itself. These are areas that the shepherd should look closely at, as they tend to be troublesome. Therefore, there is often great potential for improvement in these areas. Interestingly, it isn't easy to separate these two groups of patterns. In fact, the first pattern in the following also appears in the preceding list! In other words, if you apply BIG PICTURE with the intent of helping yourself shepherd, you are automatically helping improve the pattern itself. Conversely, if you apply it with the goal of helping improve the pattern, you are making life easier for yourself as a shepherd. This is true to a lesser extent with many of the patterns in this language.
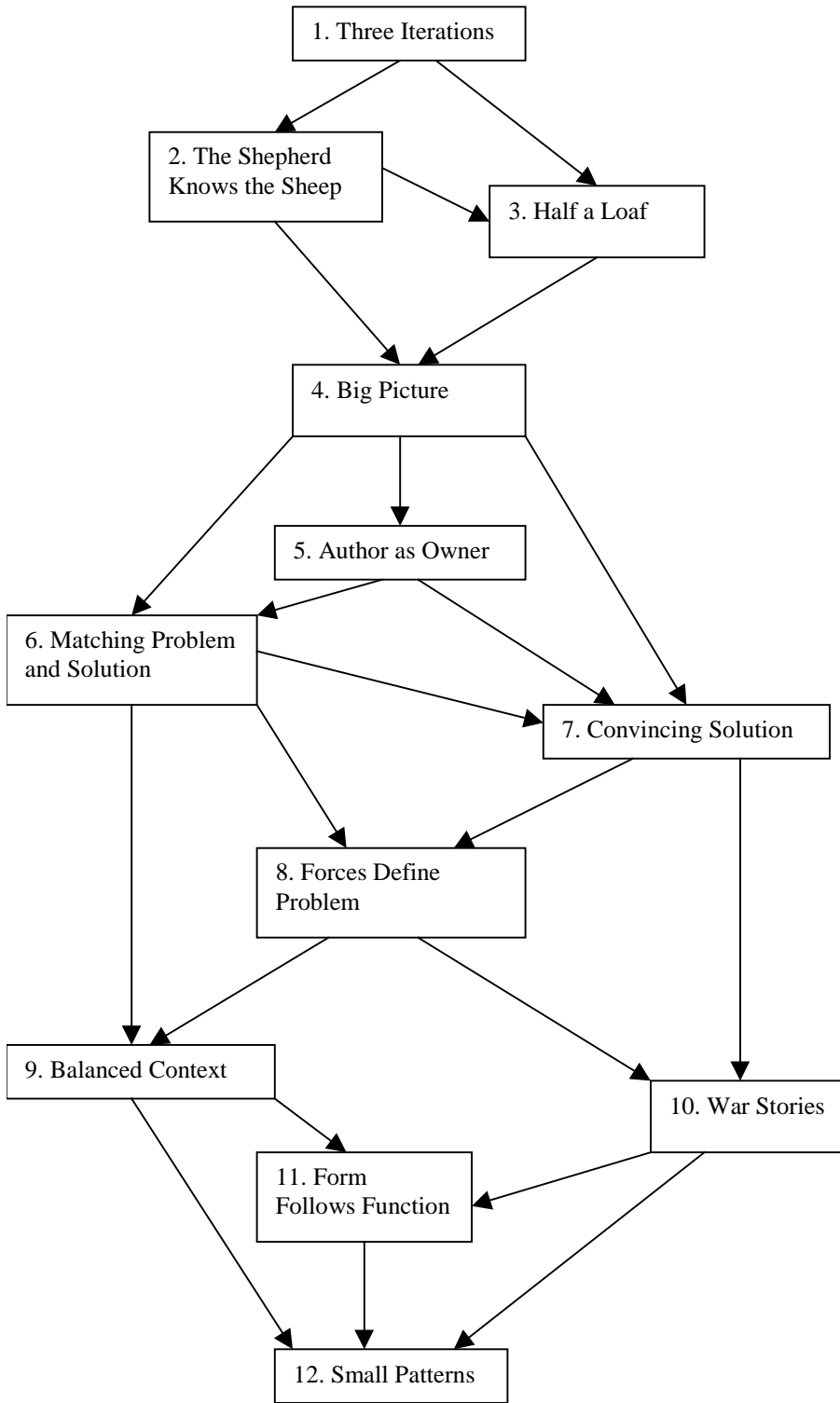
4. *Big Picture:* How to make the essence of a pattern immediately clear to the reader.
6. *Matching Problem and Solution:* How to ensure that the pattern really is pattern-ish.
7. *Convincing Solution:* How to make the pattern believable.
8. *Forces Define Problem:* How to strengthen the problem.
9. *Balanced Context:* How to help get the pattern at the right scope.
10. *War Stories:* How to help the pattern flow.
11. *Form Follows Function:* How to put a new form into a pattern.
12. *Small Patterns:* How to keep patterns easily digestable.

These patterns look more at content. They relate to patterns in "A Pattern Language for Pattern Writing."[2] These will be referred to as Pattern Writing Patterns.

During the course of shepherding, one tends to use more of the tactical patterns initially, but then tends to mix the patterns of the two groups. Here is a picture of all the patterns, showing the rough order in which they might be applied. An arrow from one pattern to another indicates that the first pattern helps set the stage for use of the second.

---

[1] Coplien, James: "A Pattern Language for Writers' Workshops," in *Pattern Languages of Program Design-4*, Addison-Welsey, Reading MA, 1999, pages to be determined.
[2] Mezaros, Gerard, and James Doble: "A Pattern Language for Pattern Writing," *in Pattern Languages of Program Design-3*, Addison-Wesley, Reading MA, 1998, pages 529-574.

```
                    ┌─────────────────────┐
                    │  1. Three Iterations │
                    └─────────────────────┘

  ┌──────────────────┐
  │ 2. The Shepherd  │
  │ Knows the Sheep  │────────┐   ┌──────────────────┐
  └──────────────────┘        │   │  3. Half a Loaf  │
                              └──▶│                  │
                                  └──────────────────┘

                    ┌─────────────────────┐
                    │   4. Big Picture     │
                    └─────────────────────┘

                    ┌─────────────────────┐
                    │  5. Author as Owner  │
                    └─────────────────────┘

  ┌──────────────────┐
  │ 6. Matching Problem │        ┌──────────────────────┐
  │ and Solution        │        │ 7. Convincing Solution│
  └──────────────────┘           └──────────────────────┘

                    ┌─────────────────────┐
                    │  8. Forces Define    │
                    │     Problem          │
                    └─────────────────────┘

  ┌──────────────────┐                    ┌──────────────────┐
  │ 9. Balanced Context │                 │  10. War Stories │
  └──────────────────┘                    └──────────────────┘

                    ┌─────────────────────┐
                    │  11. Form            │
                    │  Follows Function    │
                    └─────────────────────┘

                    ┌─────────────────────┐
                    │  12. Small Patterns  │
                    └─────────────────────┘
```

The Patterns

1. Three Iterations

When you receive a pattern to shepherd, you have an unknown amount of work ahead of you. Pattern languages have different levels of maturity and quality, not to mention size. The amount and type of work varies widely, and it's pretty hard to predict.

It can help if you know the author. You may have a better idea of what to expect from the pattern. However, different works from the same author will require different amounts of shepherding, sometimes widely different amounts.

<p style="text-align:center">∗   ∗   ∗</p>

All too often, the author receives poor feedback from the shepherd: it comes late, it is incomplete and/or superficial. The author has no chance to bounce corrections off the shepherd, or even to find out what the shepherd really meant!

Actually, most patterns need a fair amount of work when they come in for shepherding. One reason for this is that the shepherd may well be the first real "outsider" to examine the work closely. And it takes someone who is removed from the pattern to really see the strengths and weaknesses of the pattern.

Most shepherds begin with the best intentions. But shepherding is a volunteer activity, and your real job gets in the way. The time slips by, and soon all you can do is to skim the patterns and dash off a hurried set of comments to the author. You feel vaguely guilty, but console yourself that the patterns did indeed look pretty good. At least they did during your quick pass through them…

Therefore:

Plan the shepherding effort to take three iterations of comments from the shepherd to the author. In order to make three iterations possible in the time generally allotted to shepherding, it is almost always necessary to start immediately. Both the shepherd and the author must be prepared to respond quickly – the shepherd must send suggestions to the author, and the author must prepare revisions of the work.

Three iterations helps you budget your time so that you have enough time to give meaningful comments to the author. In addition, it helps you avoid the tendency to dump all your comments on the author three days before the deadline. The author gets manageable chunks of information, and can respond to them better. "Half a Loaf" expands on this point.

If you are a natural procrastinator, it may be especially difficult to do this. In order to help you motivate yourself, immediately tell the author when to expect each batch of comments. External commitment is a wonderful goad.

Some works may need only two iterations, while others might require more than three sets of comments. If the work is in good shape after one or two iterations, you can rejoice in the gift of unexpected time. However, it is the rare paper that cannot benefit from three sets of comments. It is much more common that a paper needs more than three iterations. If you can fit an additional iteration into the schedule, feel free to do so. But after three iterations, the work tends to become stale to the shepherd as well; you have moved "too close to the coalface"[3], and you find yourself repeating the same suggestions to the author. After three iterations, the effectiveness of one shepherd tends to decrease. Regardless of the state of the work, three seems to be the best number of iterations.

---

[3] Thanks to Alan O'Callaghan for this British metaphor.

A typical consequence is that shepherding takes more effort, but the quality of the shepherding will be much higher. As a shepherd, you must plan for this. Recognize that the time you spend will be well worth it to the author.

<div align="center">＊   ＊   ＊</div>

This pattern sets the stage for the shepherding session, yet it cannot succeed by itself. In order to effectively budget one's time, one must also budget the information flow; HALF A LOAF makes this happen. In addition, this pattern assumes an implicit contract of cooperation between the shepherd and the sheep. Use THE SHEPHERD KNOWS THE SHEEP to make this happen.

Thanks to Jim Coplien and Dirk Riehle.

2. The Shepherd Knows the Sheep

Now that you have the pattern in hand and are ready to begin shepherding, what is the first thing you do? You have planned out your schedule to accommodate THREE ITERATIONS. However, you realize that there is no way to assure that sheep will pay attention to your comments.

<div align="center">∗   ∗   ∗</div>

Because the shepherd is essentially a critic, there is a natural barrier between the shepherd and the sheep – the sheep feels vulnerable and defensive. This attitude can hinder effective communication, and can even foster a tendency for the sheep to ignore comments from the shepherd.

On the one hand, you might feel inclined to give lots of comments to the author in order to be the most help. But it is discouraging if the author ignores most of your comments. So you might be tempted to gloss over the difficult items, and give only cosmetic suggestions, since they are non-threatening.

Most shepherding is done via email. While email is nearly ideal for shepherding, it is detached and somewhat anonymous. This makes it easy to ignore.

Since most shepherds are also authors, one can follow the Golden Rule – I should respond quickly and positively to my shepherd, and perhaps my author will be responsive to me. But positive example only goes so far; there are no guarantees. We are dealing with other people, and have no real power over them.

Therefore:

Start immediately to build a relationship with the author, and maintain it throughout the shepherding. Contact the author right away, even before reading the patterns, and tell the author about yourself. Tell the author what to expect from you (such as three iterations), and establish what you expect from them in return. Above all, follow through on your commitments!

It is important to make the shepherding personal. The author needs to feel like they know you and can trust you. It helps if they know that you have been a sheep yourself, and know that to expect. You are, in essence, establishing a *Safe Setting*[4] for the author.

Early contact is important. The author will have been informed of your stewardship, and they will be expecting to hear from you. Establishing contact immediately shows that you care. The author will be more willing to listen to your comments. This embodies the old adage, "They don't care how much you know until they know how much you care."

You can maintain the personal relationship over time through the manner in which you give feedback. Don't forget to give positive reinforcement as well as suggestions for improvement. Try to give *Positive Feedback First*[5], and finish up each group of suggestions with *Positive Closure*[6].

An important aspect of a relationship is that both parties understand what the other expects. Establish ground rules which encompass what you will do and what you expect the author to do. Include, for

---

[4] Writers' Workshop Pattern *Safe Setting*: How to make feedback more open and effective by raising the comfort level, particularly for the author.
[5] Writers' Workshop Pattern *Positive Feedback First*: How to give the gathering a supportive tone, and to start with feedback that will put the author in a receptive mood.
[6] Writers' Workshop Pattern *Positive Closure*: How to leave the author with a positive feeling at the end of the feedback.

example, dates of trips or events that could prevent you from responding quickly. Additionally, you could mention your preferences for email etiquette, if that is important to you.

*　　*　　*

If you do this, both shepherd and author will be edified by the process. You will build relationships that last beyond the shepherding period, as you build a *Community of Trust*[7]. On one occasion, I worked extensively with an author on his paper. At the following PLoP, the author, knowing of my love of folk music, gave me a CD of folk music from his country as a thank you.

---

[7] Writers' Workshop Pattern *Community of Trust*: How to help authors feel that the experience will help them, rather than tear them down.

3. Half a Loaf

Shepherding is now underway. By using THREE ITERATIONS, you have established an approximate schedule, and you and the author understand what each expects of the other (THE SHEPHERD KNOWS THE SHEEP). Now the author is ready to receive your first batch of comments.

✳   ✳   ✳

In the zeal to help the author, the shepherd often sends every possible correction to the author all at once. This overwhelms the author and disrupts the schedule.

It's natural to want to send a comprehensive list of comments to the author all at once. But you can't possibly comment on every little thing and still maintain three iterations; there just isn't time. Even if you were able to do so, the author couldn't respond in time. After all, the job of revision is even bigger than the job of criticism!

Not all corrections are created equal. Some are absolutely critical; others are trivial. But if the author receives all the comments at once, it is difficult to know which are the important ones. The author may spend excessive time making minor changes, and not have time for the big ones. This is a natural tendency: it's easy to correct grammar and spelling, and we tend to like to do the easy things first.

Therefore:

Send the author small bites of comments. Start with the biggest issues, and work to the most trivial. If you run out of time but haven't assembled all the comments you wish to make, send what you have. The author can start on whatever comments you send. Half a loaf is better than none.

Comments, unlike software, don't need to be complete before they are released. In fact, they don't even need to be debugged! The author will be able to make sense of them. And if the author can't, you can expect some e-mail asking for clarification.

The schedule is more important than the completeness of the comments. For one thing, if you miss the time, you put an iteration of comments in jeopardy, and they can't be made up. For another thing, the author can begin with what you send. If necessary, you can always complete your comments a few days after you send the bulk of them.

✳   ✳   ✳

Now you are off and running. Well, maybe. There is still this gnawing question: If you are supposed to begin with the most important issues, how do you know what they are? Where do you start? The answer is found in BIG PICTURE.

Thanks to David DeLano, Jutta Eckstein, and Dirk Riehle.

4. Big Picture

Now that you are ready to give a small dose of feedback (see HALF A LOAF), the task before you is to understand the pattern and find the most important issues to highlight first. But this is easier said than done.

$$* \quad * \quad *$$

It's often hard to know where to start shepherding. You need to understand the pattern in order to make effective suggestions, but draft patterns are often pretty arcane. Patterns that are small and concise at this stage of maturity often lack substance. On the other hand, large patterns often have extraneous details that obscure the main ideas.

Upon first reading, one can find many areas for improvement in a draft pattern. But time is limited. So it's critical to call out the most important issues first, and get them straight. Furthermore, some of the minor issues may change, or even take care of themselves as you tackle the big issues. But it isn't always clear what the most important issues are.

It's not uncommon for a draft pattern to be pretty muddy – it's hard to figure out what the pattern is all about. There may be a great pattern struggling to get out, but it takes work to unearth it. To put it bluntly, a lot of engineers are poor writers!

Of course, the shepherd's initial experience to a pattern will likely mirror the experience of the general populace. First impressions tend to stick. So the way a pattern is presented can make all the difference.

Therefore:

Start by reading the problem and solution of the pattern to get the main idea, and to give feedback on the problem and solution first. By themselves, the problem and solution should give the big picture of the pattern.

You start here for three reasons: First, it gives you an idea of the pattern very quickly, and helps frame your subsequent study of the pattern. Second, it helps you see how (or if!) the author understands the pattern. Third, the essence of a pattern should be easy to pick out of the pattern itself. The reader must be able to find what the pattern is about relatively easily. By the way, it's best to give your first impression feedback very soon after you read the big picture, similar to *Reading Just Before Reviewing*[8].

A good place for this kernel information is at the beginning of the problem and solution sections. Look at the first one or two sentences in each section. You should be able to understand the basic idea of the pattern upon first reading. This is evocative of *Single-Pass Readable Pattern*[9]. If it isn't, you can ask the author to explain what the pattern is all about as succinctly as possible, and incorporate that into the pattern. You might have the author use a 'patlet' form, and work from the patlet for a while.

Note that some patterns (like these) don't have sections explicitly called "Problem" and "Solution." Nevertheless, they should be easy to find (see *Findable Sections*[10].) In these patterns, for example,

---

[8] Writers' Workshop Pattern *Reading Just Before Reviewing*: How a reviewer should avoid over-preparing or under-preparing for the workshop.
[9] Pattern Writing Pattern: *Single-Pass Readable Pattern*: How to make it easy for the reader to understand the gist of the pattern on first reading.
[10] Pattern Writing Pattern *Findable Sections*: How to make key information easy to find.

problems and solutions are in bold. Note that the problem and solution are the key elements of *Mandatory Elements Present*[11].

<p style="text-align:center">✲   ✲   ✲</p>

Shepherding tends to be more effective when you start with the BIG PICTURE. It helps get the shepherd and the author on the same page. It can avoid costly (in time) misunderstandings, and it helps place the focus on the most important issues first. (See MATCHING PROBLEM AND SOLUTION.)

An interesting parallel pattern for reviewing papers is *Evaluate Papers Fast*,[12] by Jens Parlsberg.

---

[11] Pattern Writing Pattern *Mandatory Elements Present*: How to make sure all the necessary information is present in a pattern.
[12] *Evaluate Papers Fast*: How to quickly get an idea of the merits of a paper to be reviewed for a conference or a journal. See http://c2.com/cgi/wiki?EvaluatePapersFast

5. Author as Owner

Shepherding is now underway. You have read through the pattern, and have probably used BIG PICTURE to help you identify the highest priority issues of the pattern. You are now ready to send your first comments to the author.

<div align="center">✳ ✳ ✳</div>

Developing a strong relationship with the author can cause the author to lean on you too much. In particular, the author may use your suggestions verbatim.

It is somewhat flattering to have the author use your words exactly, but it has negative consequences. First of all, you aren't the expert, so your comments may not be completely accurate. After all, the author should be the expert, not the shepherd.

A second problem is that it makes you, in a small way, a part owner of the work. This is usually inappropriate, and it's a burden that most shepherds don't want.

Perhaps the biggest problem is that the author doesn't learn by simply copying your suggestions. Grappling with technical issues as well as stylistic pattern problems is one of the best ways to grow; the author shouldn't be deprived of this opportunity.

However, time pressure encourages the author to simply incorporate the comments into the pattern without giving them much thought.

Therefore:

Establish the author as the clear owner of the work by not spoon-feeding answers to the author. Instead, ask questions in order to nudge the author in the right direction. Phrase much of your feedback as thought-provoking questions.

It's often hard to turn your criticisms into questions that will help the author come up with the answers. Think about "what", "how" and "why" questions, such as, "What problem are you solving?" or, "How does the solution solve the problem?" or, "Why is this force important?" You might even use other journalistic "w" questions, such as, "When does this solution not work?"

Questions force the author to think about the problem, and to come up with the right words (and concepts!), rather than taking your words without much thought. The author will therefore learn about the topic and patterns in general. Meanwhile, you become the invisible hand behind the work, without becoming overly involved in it.

<div align="center">✳ ✳ ✳</div>

Note that this approach takes more work than simply tossing off a bunch of suggestions to the author. As a shepherd, you need to think carefully about what questions to ask the author. It is also more work for the author. But this extra time and effort pays off in a more consistent, higher quality pattern.

A side effect of questions is that your comments are not direct attacks on the work, and thus are less likely to offend the author.

6. Matching Problem and Solution

Once you are on your way with the BIG PICTURE, you need to know what to do with it. Most patterns that are being shepherded need some work to make the BIG PICTURE hold together.

∗   ∗   ∗

Often, an immature pattern doesn't hang together well. It may feel like a solution looking for a problem, or maybe the purpose of the pattern isn't clear.

Patterns are not written front to back. Authors bounce around in the patterns as they write them. So the "Big Picture" of the pattern – the problem and solution – are often written at different times. But the reader reads the pattern from front to back. This often leads to a Big Picture that is distorted or fuzzy.

We tend to focus on solutions; indeed patterns are about solutions. This focus leads to strong solutions with problem statements that are little more than afterthoughts. But we need strong problem statements to know when we might need a pattern.

Some patterns are pretty complex, so they are naturally hard to understand. They require a great deal of detail, yet this detail can obscure the meaning of the pattern. This is exacerbated by the fact that the author, who is an expert on the topic, tends to focus on the details; the gist of the pattern is now so obvious (to the author) that it gets little attention. Yet it isn't obvious to the reader.

Therefore:

Read the problem and solution together to make sure they match. The solution must address the whole problem, but not more than the problem. Identify which of the two sections is weak with respect to the other, and strengthen it. In general, work on the solution first, then the problem.

This is all about balance: the problem and solution must fit together in order for the pattern to feel whole. One aspect of this is the breadth of the pattern. The problem and solution must cover the same ground. A broad problem requires a general solution, and a narrow problem is covered by a specific solution. Another issue is depth: solution and problem should match here as well. If a solution addresses one issue in depth, and that issue is germane to the solution, it must be reflected in the problem, or perhaps the context. (The same goes for depth in the problem, although this is uncommon in practice.)

You may still be on the first pass over the pattern at this point, so it still may be helpful to read just the problem and solution, and skip the other sections. This will help give you a sense of the pattern's balance. However, you will need to begin to bring in the surrounding information of forces, context, and implementation to fill in some holes.

Because we write patterns based on our experience with the solutions, we tend to write the solution of a pattern first. At least, that's what we have in mind first; the solution motivates us to write the pattern. So the solution is usually more mature than the problem.   This is why we start with the solution.

∗   ∗   ∗

This pattern helps you see what part of a pattern might need work. It also helps the author understand the interaction of problems and solutions. But beyond identification, you need specific action to fix the solution and the problem. See CONVINCING SOLUTION and FORCES DEFINE PROBLEM for the next steps.

7. Convincing Solution (The "Aha" Effect)

By now, the pattern is taking shape. It has a solid BIG PICTURE. The problem and the solution are beginning to work together (MATCHING PROBLEM AND SOLUTION). As the shepherd, you may have helped the author apply these patterns, but in many cases, the pattern already has these characteristics when it comes to you. On the surface, the pattern looks good.

But something about the pattern just doesn't feel right. In particular, you are skeptical of the solution; you aren't convinced that it works. Perhaps the problem is that the pattern does not show evidence that the pattern has been successfully used. Or maybe the pattern exhibits the "World Peace" syndrome; the solution is easy to say, but hard to do. Once in a while, the solution actually seems wrong.

<p style="text-align:center">✳   ✳   ✳</p>

Sometimes the solution is really weak; it just doesn't do anything for you.

Patterns should capture proven knowledge. But sometimes authors write up a single-instance solution as a pattern. In rare cases, they write up a proposed solution; one that hasn't even worked once (these tend to look like "World Peace"). At writers' workshops, we tend to believe the author, so it is really up to the shepherd to catch these cases to avert embarrassment later.

The author is often so close to the pattern that some important aspects of the solution seem obvious, even trivial. So they aren't mentioned. But this can cause the credibility of the pattern to suffer. This can make the pattern look like a "World Peace" solution, because information that would help us apply the pattern is missing. So we wonder whether the author has really successfully used the pattern.

Some people are enamored with "anti-patterns", which describe a common problem, and then advise against the destructive behavior. But they don't tell what to do instead. Patterns should convey information and insight about what works. Most of us already know dozens of ways to fail.

Therefore:

Look for a solution that is strong, even compelling. Ideally, the solution should have the "Aha" effect. If it isn't, ask direct, pointed questions to bring this out. For example, tell the author that you are not convinced that the solution works; please convince me. Ask how to implement it, or ask where the author has seen the pattern in practice.

Watch for tip-offs, such as the word "should", or little evidence of known uses. Above all, trust your own instinct. If something sounds too good to be true, it probably is.

Solutions that are extremely broad tend to lack credibility. Help the author make the solution narrower and deeper by asking for specific examples of usage. You can also present counterexamples, and narrow the solution space by exclusion.

<p style="text-align:center">✳   ✳   ✳</p>

This is a direct challenge to the author. There is danger of damaging relationships here; that's why it's important to have established a relationship with the author (see THE SHEPHERD KNOWS THE SHEEP.) However, you must not back away. Being openly skeptical now can save the author no end of trouble later.

After (and only after) the solution is in reasonably good shape, you can move on to the problem with FORCES DEFINE PROBLEM.

8. Forces Define Problem

Generally, after you have worked with the author on a CONVINCING SOLUTION, the solution of the pattern is in good shape. But that was the easy part. Because patterns start as solutions seen over and over, the solution is the strongest part of an emerging pattern. But what is the nature of the problem that is solves?

*　　*　　*

Many patterns have poor problem statements. Sometimes they are too vague. More often, though, they point directly to the solution and only the solution. In some cases, you can't even find a statement of a problem.

The problem is the key to the pattern; if the solution is the heart of the pattern, the problem is its soul. A carefully crafted problem statement can be a boon to a person who is looking for a solution to that problem. However, many authors don't understand this, and instead focus exclusively on the solution.

Problem statements are hard. We tend to be solution-oriented; once we've solved a problem we think only of the solution. Because of this, we usually begin by writing the solution, and then write the problem as an afterthought. This leads to problem statements that presuppose the solution. In extreme cases, the problem statement is nothing but a restatement of the solution.

Because the problem comes first, authors occasionally write the problem before they write the solution. But this is also problematical. The temptation is to write a problem down even before the solution is well understood. Unless the author has thought through the solution in detail, such problem statements tend to be vague and overly broad.

Problems are complex. They have various aspects that may suggest different approaches to solutions, not all of which may work. Furthermore, the external manifestation of a problem is often not the problem itself; measles is a bodily illness that presents itself as small red dots on the skin.

Therefore:

The problem statement should describe the visible manifestation of something wrong. Then the forces give substance to the problem, and insight into what is behind the symptoms. The author should iterate between the forces and the problem statement to improve both.

Note that the problem statement in this pattern points out that problem statements tend to be weak (which is visible), and the next paragraphs (forces) point out the reasons behind it. Note also that in this particular form, the forces are not explicitly labeled. As a shepherd, you may have to hunt for the forces in the pattern.

If the author has a good start on the problem, but the forces are weak, ask what makes the problem difficult. Also ask what is the real problem that lurks behind the symptoms described in the problem. Make sure, though, that the problem statement focuses on the external symptoms, for this is what the reader will encounter and be able to relate to.

If there are forces but a problem is not articulated, ask the author to summarize the forces in a single statement that describes the problem, and go from there. If the pattern is weak in both forces and problem, start by asking something like, "So what problem are you trying to solve," or, "What is wrong with the world that this pattern solves?"

If you can't find the forces at all, it's time to apply *Visible Forces*[13].

Be particularly wary of problem statements that are questions, as they are easy to abuse. If the problem reads something like "How do you do X?", it very often presupposes the solution. Read the forces to see whether there is a real problem lurking in the forces.

<p style="text-align:center">✻　　✻　　✻</p>

You may find that you spend the bulk of the shepherding effort working on the problem. This is appropriate. Once the problem and solution are in good shape, the pattern emerges. I have seen many alleged patterns where there was indeed a pattern struggling to get out. The pattern became apparent as the problem crystallized.

---

[13] Pattern Writing Pattern *Visible Forces*: How to make sure the reader understands the choice of solution.

9. Balanced Context

As the pattern takes shape through MATCHING PROBLEM AND SOLUTION, CONVINCING SOLUTION, and FORCES DEFINE PROBLEM, it becomes more clear where a pattern can and can not be used. The problem is well understood. The solution is now better defined; this puts natural limits on its applicability. But all too often, we don't think about the effects of the solution, we just assume that it solves the problem, and that's that. But such an assumption is in reality naïve. Any action, including applying a perfectly good solution, has consequences. A pattern owes it to its users to explain them.

<p align="center">✳   ✳   ✳</p>

The pattern suffers from the "And They Lived Happily Ever After" syndrome: if you apply this pattern, all will be right with the world. Or it may have the "Good For What Ails You" syndrome: you can use this pattern for whatever problem you happen to have at the moment. Or it has both.

These two syndromes are really the same problem: the context of the pattern is not well described. On the one hand, the context for applying the pattern may be inadequate, or too broad. On the other hand, the consequences, or the context that results from applying the pattern is ignored. Both are easy to forget, particularly for the author, who is so close to the pattern.

In many cases, the pattern tries to be all things to all people. The author has received comments from others about how the pattern might be applied in this or that situation, and the author tries to expand the pattern to cover the new areas. So the pattern grows, and tends to lose focus.

If it is easy to lose focus about where a pattern might be applied, it's even easier to lose focus about what the application of the pattern does to the world. One major reason is that the consequences of a pattern are often added last, almost as an afterthought. It can get neglected in the heat of battle.

Therefore:

Look at the context before and after the solution to see how the world changes as a result of applying the pattern. Compare the beginning and the resulting contexts to each other. Also compare the beginning context to the problem, and the resulting context to the solution.

The beginning context lays the groundwork, setting the stage for what will come, namely the problem. The resulting context should explain how the world has changed through the application of the solution. The ending context should also show how the forces have been balanced.

Compare the beginning and the resulting contexts. You can list them, and see whether the results account for the entire beginning context. The overly rosy resulting context statements are usually pretty short on specifics, and so do not show how the context has changed or how the forces are balanced. So use them to guide you as you ask the author how each force has been addressed. Ask for specifics and examples. WAR STORIES may help.

Context is difficult to get one's arms around, so it is often impossible or impractical to state the context with a great deal of rigor. So don't be too hard on the pattern writer. But do watch for obvious traps: an overly broad beginning context, or a very optimistic resulting context must be tightened.

<p align="center">✳   ✳   ✳</p>

This helps keep the patterns crisp. It can also help keep the patterns focused on what is important, and make them more useable out of the box. The user will know better what to expect, and won't be built up and then disillusioned by grandiose expectations.

10. War Stories

The pattern should be improving nicely by now. The problem and solution should be clear, and one should be able to understand where to apply the pattern. But it's not. It's still difficult to understand the pattern. Furthermore, and maybe because of that, the pattern is boring.

∗   ∗   ∗

The pattern is unclear, or sterile, and no matter what you suggest, it's not getting any better.

The pattern writer is often faced with requests to add this or that to the pattern to make it more broadly applicable, easier to use, etc. But if the author adds everything in, the result is huge, unwieldy, and obtuse. In addition, as patterns grow, they tend to become more abstract, and less approachable. Yet the information requested is important.

The author is very close to the pattern, and has experienced it first hand. So the pattern is perfectly clear to the author. But it isn't to many readers. Among other things, the author has unwritten knowledge which seems obvious, but isn't to the outside world. Sometimes there is so much missing that the shepherd knows that the pattern isn't clear, but can't even figure out what to ask to gain the necessary clarification.

Therefore:

Ask the author to relate real-life experiences ("war stories") to clarify the pattern.

War Stories can be extremely powerful instruments to show what a pattern is all about, and how one can use it.

One way to use a war story is to make it a *Running Example*[14].

An important aspect of a War Story is that it be appropriate for the intended audience. See *Clear Target Audience*[15] for help.

One way to draw out relevant stories from the author is to ask how the pattern came about. I was mentoring two people at a EuroPLoP, and the pattern was just hard to grasp. We struggled together for a while, then I asked them to tell me how they came across the pattern. They immediately lit up, and talked animatedly about their use of the pattern. All of a sudden, their pattern made sense, so I then told them to write down what they had told me. The improvement to the pattern was dramatic.

∗   ∗   ∗

War stories can really make a pattern come alive. They not only help illustrate principles and reinforce the credibility of the pattern, they make it more interesting as well. The price for this is that the pattern gets longer. And two or more stories, particularly running examples, in the same pattern can feel overdone.

Note that some cultures tend to discourage the sharing of personal experiences. Where this is the case, this pattern may be inappropriate.

---

[14] Pattern Writing Pattern *Running Example*: How to make it easier for the reader to put the pattern into practice by using a single example through several patterns.
[15] Pattern Writing Pattern *Clear Target Audience*: How to make sure the pattern reaches the intended audience by clearly identifying a primary target audience.

11. Form Follows Function

(Or: Functional Form, Fitting Form)

Applying patterns such as FORCES DEFINE PROBLEM and BALANCED CONTEXT often cause large chunks to be added to a pattern. Sometimes they don't fit well with what is already there. In addition, sometimes the material in the various sections of the pattern seems somewhat contrived. This may be apparent early on, but might not become glaringly obvious until after one has wrestled with a MATCHING PROBLEM AND SOLUTION, and forces and context.

<div align="center">✳   ✳   ✳</div>

Sometimes the pattern and the form the author picked just don't go together.

Some authors have limited exposure to patterns, and therefore may know only one or two pattern forms. For example, many people's first experience with patterns is the GoF book (Design Patterns: Elements of Objected-Oriented Software Design), which uses a different pattern form than many patterns in the PLoPD (Pattern Languages of Program Design) series. You stick with what you know.

However, some patterns just work better in one format than another. And the author may have a much easier time with a different form. I remember once writing a group of patterns in a particular form. I struggled with them until I started over, using a different form. Then the patterns almost wrote themselves.

But a form change can be major. The author may be (understandably) reluctant to make such a big change, especially when there is little time in which to rework the patterns. It's hard to change horses in the middle of the stream.

Therefore:

Find a form that fits the pattern, and slide it in, one part at a time.

Ask the author to provide some information, and then suggest that it might work well as its own section with an appropriate name. For example, if the target pattern form has a separate problem statement, you might draw out the problem as described earlier, and suggest that it be set off with a "Problem" heading. You can eliminate sections by suggestions to remove them, or combine them with other sections.

The goal, of course, is to make the form serve the pattern, rather than the other way around. As you gradually introduce the desired form, the author should see how the new form is useful.

<div align="center">✳   ✳   ✳</div>

Of course, as all the other patterns, this pattern will not force the author to write a certain way. It's still up to the author. At the very least, though, the author will be exposed to new pattern forms.

Note that this pattern works very closely with SMALL PATTERNS, which follows.

12. Small Patterns

(Or: Small is Beautiful)

Most of the previous patterns focus on enhancing and clarifying the substance of a pattern. This means adding material to the pattern. And sometimes that material can be substantial.

∗   ∗   ∗

Revising a pattern, especially with input from others (such as shepherds), naturally causes it to grow into a behemoth.

It is vital that a pattern have enough information to be usable. However, if the pattern is too big, the readers get lost. Brevity is the soul of wit.

As the author adds to the pattern, the new information may well overlap with the old parts of the pattern. But that may not be initially obvious. On top of that, the author is understandably not eager to rip anything out. After all, he wrote everything for a reason.

One might consider trying to keep the pattern small by ongoing pruning efforts. But it often isn't clear what information is needed in a pattern until it has been worked quite a bit. But by then, the author may be reluctant to change the pattern any more at all, particularly if it involves removing stuff.

Therefore:

Allow the pattern to grow at first, then cut the pattern down. Identify extraneous chunks of the pattern and remove them.

Although one can usually tighten the prose of a pattern, redaction yields only minor reductions. You need to find whole parts of the pattern that are unnecessary, and take them out. For each candidate section, ask what it adds to the pattern, and whether it is covered elsewhere. This helps make it clear to the author which parts of the pattern should be removed. This is often done together with FORM FOLLOWS FUNCTION.

Hypertext supports detail that can be hidden on first reading and brought forth if desired.

If the pattern is very large, it may actually be several patterns. You might suggest that the author consider whether the pattern should be broken into multiple smaller patterns. A tale-tell sign is when a pattern contains multiple solutions; a single pattern should contain one problem, one context, and one solution.

∗   ∗   ∗

The resulting pattern is smaller and cleaner than before, but it is almost always larger than it was at the beginning of shepherding. That's usually all right; if it isn't, you can make multiple passes on the pattern. Cutting can be a difficult process, but it is still easier with prose than it is with source code.

Candidate Patlets

The patterns community is fortunate to have a large body of experienced shepherds. I polled the current shepherds for the 1999 PloP conferences, and asked for their shepherding experiences. Their responses tended to confirm and amplify the previous patterns, and suggested several new patterns. The following patterns are from them; names of the originators are given at the end of each. I have included my own comments in italics.

This is a work in progress. Please provide feedback, either about these candidates, or about new candidate patterns. Thanks.

Patlets Drive Discussion

Pattern sizes are inconsistent, irregular, or awkward.
Therefore, have the author write patlets for the patterns, and at first, focus the discussion on the patlets.
(*This is important, but is it just a part of Big Picture?*)
(Ward Cunningham)

Details Can Wait

The pattern has many annoying problems with grammar, spelling, sentence structure, etc.
Therefore, ignore them until late, even if they really get in the way. When you are doing major remodeling, you leave the wallpaper and trim until last; otherwise you will keep ripping out the trim you just put in. (*This probably should be promoted to full patternhood*.) (David DeLano, William Opdyke)

Reserve Judgement

The work stinks.
Therefore, do shepherding first; separate shepherding from evaluation. At the end, and not before, read it for evaluation whether to accept. (*It's extremely important to separate shepherding and judgement. Is it a pattern? If so, where does it fit into this language?*) (David DeLano)

Form Proves Pattern

The work isn't a pattern, it is a retread of a paper rejected from some other conference.
Therefore, ask the author to put it into a pattern form (then proceed to Reserve Judgement…) (William Opdyke) (*Is this part of Form Follows Function, or maybe Matching Problem and Solution? To me, this feels almost backward. You should establish its patternhood, probably with Matching Problem and Solution, and then worry about the form. I'm leaning away from this one*.)

Domain Expertise in Shepherds

You find that you are totally lost in the pattern.
Therefore, select patterns in domains you know. (Implies that the program committee allow shepherds to select their own patterns.) (*I think this pattern is very important. But it doesn't seem to fit in this language, but in some closely related language*.) (Antonio Rito Silva, Jim Coplien, Linda Rising)
Common Variation: English is a domain. Papers written by non-native English speakers can profit by having a shepherd who speaks English natively or very fluently. (Andreas Rüping, William Opdyke)

A Place in the World

The pattern doesn't acknowledge related work.

Therefore, Ask how the pattern relates to (specific) previous work. Often: ask how it differs from a specific previous work. This is particularly useful where a pattern seems to be a minor variation on an existing pattern. (*The problem may be wrong. Perhaps the problem is that a pattern is a minor variant of an existing pattern. Then ask how it differs. Is this too obvious? Quite possibly, but I have used it several times.*) (Mark Bradac, Linda Rising)

Joint Self-Criticism (The Barber's Haircut)

A pattern written by a committee looks that way.

Therefore, ask each author to read the whole work to "even out the writing." It keeps you as shepherd from all that work, and keeps you from inadvertently favoring one writer over another. (*When you go to the barber, pick the barber with the worst haircut. Remember that they cut each others' hair.*) (Linda Rising.)

Interlaced Comments

Sometimes you just can't get through to the author.

Therefore, embed your comments in the document itself, so that the author can see exactly what you mean. (*Is this a pattern, or is it just a simple rule? It was originally in the language, and I pulled it out for the moment.*) (Neil Harrison and others)

Epilogue

On the surface, these patterns are all about how to help someone improve their patterns. Yes, that is what they are all about. But they are deeper than that. For the most part, these patterns are all about the things that make patterns good. For example, War Stories is all about making the pattern more concrete and approachable. Matching Problem and Solution helps the shepherd, and therefore the author, focus on the essence of the pattern – the solution and what problem it solves. Many of the patterns seek to improve the quality of the problem. If the solution is the heart of a pattern, the problem is its soul. In my experience, the problem statements are typically weak, so the patterns can directly influence the quality of the pattern.

But one can go still deeper. These patterns can help us solve problems more effectively; they can guide us in how we approach a problem; they can help us think. For example, Big Picture and Matching Problem and Solution can help us get started right. I have been in several meetings where my friend and colleague, Jim Coplien, has asked, "Wait a minute! What problem are you solving?" This question helps us focus on what we are really trying to accomplish, and steers us away from ineffective actions. In a like manner, all these patterns can help us, not only in shepherding, but in general problem solving and in relationships with others.

Acknowledgements