

How to write a pattern?

A rough guide for first-time pattern authors

Tim Wellhausen

kontakt@tim-wellhausen.de
<http://www.tim-wellhausen.de>

Andreas Fießler

patterns@fiesser.de
<http://patterns.fiesser.de>

November 30, 2011

Introduction

Pattern writing is a creative, iterative process. For new pattern authors, however, it is often not obvious where to start and how to proceed. Although well-written patterns typically follow a strict format, the number of available formats push first-time authors right at the start to choose a specific format without knowing its advantages and liabilities. Furthermore, new pattern authors are often not aware that the order in which a pattern is published typically is not the best order in which to write the pattern.

Done the right way, writing a pattern often creates unforeseen insight into a domain one was sure to know very well. The pattern structure encourages you to think more deeply about the actual problem that the pattern solves and what consequences arise by applying the solution. And by following an iterative approach you gain new insights in each cycle.

This paper aims at smoothing the learning curve of pattern writing by proposing a step-by-step guide that enables you to write a pattern in a simple but complete format. The goal is to enable you to submit your pattern to a pattern conference.

The paper is divided into two parts. Part I is a warming-up. It presents the transcript of a conversation about a pattern and shows how knowledge of a fascinating solution slowly transforms into a pattern. Part II contains the actual step-by-step guide how to write a pattern.

Part I: A conversation about a pattern

In order to demonstrate the reasoning behind writing a pattern, Tim and Andi will talk about a well-known solution while you can listen like a fly on the wall.

Andi: Hi Tim! You said you've been thinking about a solution that you'd like to write up as a pattern. Shall we try to do that now?

Tim: Yes, let's do that! What I have in mind is a lock. I guess you think you know everything there is to know about a lock. Let's see what aspects we can find!

Andi: A lock? I see. What kind of lock do you mean? A padlock?

Tim: Actually I had a door lock in mind. Almost every door has one. Why is that?

Andi: Of course many doors have locks. What is so special about it?

Tim: A lock allows you to protect some valuables against unauthorized access.

Andi: I see what you mean. But is a lock complete without a key?

Tim: You are right. Only the matching key opens the lock. You can easily take the key with you so that you can get access to your valuables at any time. Furthermore, you can give the key to other people so they can also get to the valuables without you being present.

Andi: That sounds indeed like a neat solution. Can you summarize the essential idea in a single sentence?

Tim: Let me try: A mechanism that uses a matching key to either allow or prevent the door from being opened.

Andi: Ok - Let's think about the problem. Why would you use a lock?

Tim: You want to protect some valuables.

Andi: Sure, but that's not the main point of this solution. Just for protection, you could also board up the door.

Tim: So we have to be more specific! We are talking about some space such as a building or a room where we would like to keep things. We want these things to be protected from any unauthorized access.

Andi: Yes, and we want to get access to the things stored there easily and at the same time make it difficult for others to get to them.

Tim: And we want to share the access to these things easily with selected other persons.

Andi: That sounds good. So what is the problem?

Tim: We want to protect things that take space from unauthorized access while at the same time we want to easily share access to these things with selected other persons?

Andi: So, in the context of a room that has a solid door as only opening, the solution would be: Store the valuables in the room and attach a lock to the door that can only be opened and closed by matching keys.

Tim: Sounds trivial to me. Shouldn't we describe in more detail how a lock works?

Andi: Tough question! In reality, there are so many different kinds of locks and keys. The deeper we dig, the more patterns we will find. Just think about the internals of a lock that you operate with the key.

Tim: No doubt the internals of a lock can be fascinating. But is that in the focus of our current pattern?

Andi: You are right, let's concentrate on our solution. We are not done yet.

Tim: So what's next?

Andi: Let's have a closer look at the problem domain. Why is the problem difficult to solve? How would the world be without locks and matching keys?

Tim: There would be no privacy or private ownership – at least in a society where some people are eager to take things away from others.

Andi: That's an important aspect of the context. I know of neighbourhoods in Utah, Scandinavia and New Zealand where no one locks their houses or cars because everyone respects the property of the others.

Tim: So if we assume that malevolent people exist, without locks and keys, we could not protect our valuables.

Andi: Think of my objection that you could just as well board up the door.

Tim: Right. We want to safely store our valuables while, at the same time, access to them should be easy, both for us and for selected other persons.

Andi: How can we distinguish our solution from similar solutions? I could think of a combination lock instead of a lock with a fitting key.

Tim: You don't want to keep any secrets in mind. And you want to be able to withdraw the permission to access your valuables from persons to which you gave the permission before. If you take the key back, that person cannot get to the locked valuables any more.

Andi: These are important forces. Are there more consequences than we already talked about? A liability comes to my mind: You cannot easily duplicate a key to give access to more people. Well, it's also an advantage: This makes it difficult for intruders to create a copy of a key. So I know that my property is safe if I know the location of all of my keys. But if I lose a key, then I need to replace the whole lock.

Tim: Good points! Let's go back to your opening question: What is the difference to a padlock?

Andi: A door lock has the advantage that you can lock it from both sides of the door. The advantage of a padlock would be that you can lock something somewhere spontaneously where a fixed lock is missing. And you can exchange it when a key gets into the wrong hands more easily.

Tim: That is indeed a liability of the door lock. How can we then extend the problem statement to incorporate that we want to be able to lock a door from both sides?

Andi: Let's narrow down the context again even if we lose some situations for which the key and lock example would be applicable. It makes life easier for us now and we can still extend the context again later on.

Tim: You are right again. What about this: You own a house and want to protect your property inside with or without you being at home yourself.

Andi: Is this the context or the problem?

Tim: The context is: You own a house. The problem is: You want to protect your belongings.

Andi: Sounds good to me. The pattern's name probably is Door Lock, isn't it?

Tim: I can't think of a better name right now.

Andi: So let's summarize our thoughts and make some notes. How do we categorize them?

Tim: For now, we should be fine to divide them into statements about the problem and the solution domain.

? Problem Domain	Solution Domain !
<ul style="list-style-type: none">• You own a house• Room has a solid door as only opening• Privacy or private ownership• Malevolent people • Protect your belongings and yourself• Block unauthorized access • Don't want to memorize secrets• Safely store your valuables• Easy access for you, difficult for others• Share access with selected other persons• Easily withdraw permission• Operate from within and outside	<ul style="list-style-type: none">• Aliases: Door Lock, Lock & Key, Selective Access• Integrate a locking mechanism into every door• Repeatably allow or prevent door from being opened• Operated by a matching key• Accessible from both sides • Duplication of keys is difficult• Change lock when a key gets into the wrong hand

Andi: Nice draft! Did we miss something important?

Tim: I'm sure we did. But it's a promising beginning. What do we need to do to transform our notes into a full pattern?

Andi: A pattern is not just an assembly of paragraphs. Let's have a look at the structure of a pattern.

Tim: Ok. Let's start by getting an overview before we do the next iteration to refine our thoughts into a pattern.

Part II: The Writer's Path

The following guide is divided into several individual steps. These steps will guide you through all sections of your upcoming pattern. The given path is one that we have experienced as an easy one without too much complexity. There may be others as well. In particular, as writing is not a one-pass linear process, feel free to step back any time to revise your earlier writings. When you are finished once, consider starting over from the beginning again.

A quick overview

Besides its name, a pattern contains at least the following five sections: Context, Problem, Forces, Solution and Consequences (positive and negative):

- The **Context** section sets the stage where the pattern takes place.
- The **Problem** section explains what the actual problem is.
- The **Forces** section describes why the problem is difficult to solve.
- The **Solution** section explains the solution in detail.
- The **Consequences** section demonstrates what happens when you apply the solution.

In most pattern publications, these sections succeed in the order just given. Unfortunately, it is very difficult to write a pattern linearly in exactly this order!

The following figure shows the pattern sections and illustrates the order in which we suggest you to write them. We assume that you have seen and used a specific solution several times. But you might not yet know precisely what problem it solves. So we suggest to start with the solution because that is what you probably know best.

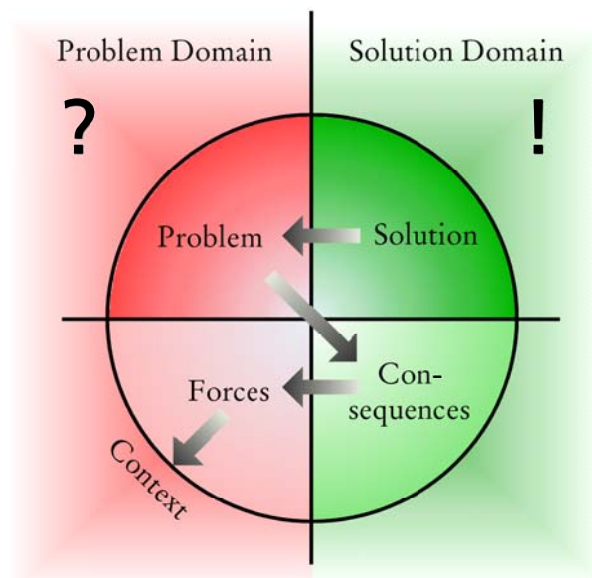


Fig. 1: Essential pattern sections and their writing order

Writing a pattern is an iterative process. Once you have a draft of each section, you will probably need to revisit them several times. The next steps explain how to do this. Every step evolves the DOOR LOCK pattern as running example.

Explore what you want to describe

We assume that you are expert in the area you are writing about and therefore have extensive knowledge about an idea, a thing or a process that fascinates you. Before anything else, have a look at it from several perspectives. Questions to answer can be:

What is your pattern all about?

- What in particular makes it something special?
- What does it include and what does it not include?
- How can it be distinguished from similar ideas, things or processes?
- What examples do you know?

Try not to write complete sentences or paragraphs, just write down notes that will help you during the next steps.

Begin with the solution

Skim over your notes and identify those aspects that describe the solution best. Tell your solution to yourself or your co-author in a few short sentences. Not any more time than you would have when you meet a friend at the traffic lights that have already been red for a while. Then write down one or two short sentences that summarize the solution. These sentences should be enough for someone familiar with your domain to understand your idea.

What is the essential core of the solution?

Collect some aliases as very short essence of your solution. There is no need to decide on a single one yet.

Skipping details is fine for now. Concentrate on the essence. What is the “Aha”- effect of your solution? When you think about the solution, you might come up with some aspects that make the solution clever. If they do not have a specific function in reaching the solution, note them on a separate sheet until you work on the Consequences section.

From the DOOR LOCK example, we have identified the following for the solution:

Solution:

Integrate a mechanism into a door that can be repeatedly operated from both sides of the door by a matching key to allow or prevent the door from being opened.

Describe the problem that leads to your solution

The transition from the solution to the problem that it solves is a difficult, yet crucial step. When you describe the problem, think about how you could impress your friend by telling that you know how to solve this problem. If you are not sure whether you know the exact problem that your solution solves, it might help to ask “Why?” and “So what?” over again.

Why is the solution relevant?

Again, restrict yourself to a short sentence that summarizes the problem.

What problem does the solution actually solve?

Try not to create a “How do you do X?” question. While it is easy to write, it is very tempting to end up with a “Do X” solution. These questions carry a high risk to end up in triviality.

To formulate a catching problem statement, think about why the world is worse without your solution applied. This phrase can be a declarative statement. It states the goal that will then be accomplished by the solution.

Check whether your problem statement really matches your solution. Read aloud both the problem statement and the solution statement. Is the solution really an answer to the problem? If they don't match, you probably need to revise at least one of them (probably both multiple times!).

Problem:

You want to control access to a building.

Continue with the consequences

Take your notes why your solution is clever and transform them into proper consequences that describe the results from applying the solution.

What happens if the solution is applied?

First think about the benefits of applying the solution. Focus on what happens if the solution is applied. If you find this hard to answer think about what happens if it is *not* applied.

Then think about the liabilities that are associated with the solution. What does a reader need to be aware of when applying the solution? What drawbacks exist? Consequences are often presented as lists of benefits and liabilities.

Consequences:

Benefit:

You can easily lock the door when you are inside your property and when you are leaving. It is hard for someone without the key to enter.

Liability:

Because the keys must be difficult to duplicate, you typically only have a limited number of them. So you cannot give access to your property to as many people in parallel as you may like.

Note the forces that make your problem difficult

Try to find reasons why it is difficult to find a solution for the given problem. It might be hard to come up with forces because you are fascinated by the solution. Nevertheless, forces are needed to make the reader aware that the problem is more difficult to solve than it might appear at first sight. They build up tension.

Why is the problem that you describe difficult to solve?

In a perfect pattern, your forces restrict the available solutions to just a single one. To find those forces, you could think about *other possible solutions* for the given problem. Aim at those forces that are resolved by your solution but are not resolved by any alternative.

Force:

You want to protect your property but you cannot be at home all the time or ask somebody else to watch your belongings. Even when you are at home you might want to protect yourself together with your belongings to have a peaceful sleep.

Match the forces with the consequences

Now it is time to think about the consequences of your solution more deeply and how they are connected to the forces.

Forces and Consequences are two sides of the same coin: A force is something that makes the problem difficult. The corresponding consequences explain how the force is resolved by the solution, for better or for worse.

A formal aspect to create a proper pattern structure is to make sure that every force is resolved by a consequence. The other way round, every consequence needs to be brought up by a corresponding force. If a force is not resolved by the solution, i.e. if the force still exists after applying the solution, it rather is part of the pattern's context.

As a start, it helps a lot to have a 1:1 relationship. To keep track of this, have a separate paragraph for every force and every consequence and give them matching paragraph titles. Don't be surprised if a force resolves into both a benefit and a liability at the same time.

Did you address every force with one or more consequences?

Force:

Storage of permission: Permission should be easy to use and reliable to store.

Consequences:

Benefit:

Storage of permission: As the key is a physical object, you can easily take it with you any time so that you can always lock and unlock the door as desired.

Liability:

Storage of permission: Because the keys must be difficult to duplicate, you typically only have a limited number of them. So you cannot give access to your property to as many people in parallel as you may like.

Context

Now you need to set up the stage for the whole pattern: its context. The context usually contains aspects and requirements that are so important that the problem may not exist outside the context but that are, at the same time, not modified by the solution. The context therefore does not change when the solution is applied. The context also helps the reader to identify patterns that might fit to their problem at a glance.

To come up with a context, think about underlying assumptions of the problem: when and how does the problem exist? If you have identified forces that make the problem difficult but that are not resolved by the solution, these forces might actually be part of the context.

Under which circumstances does the problem appear?

Context:

You are the owner of a house in an area where you cannot leave your property freely accessible to all people because this might attract thieves. The building can be accessed through DOORS.

Name

Every pattern needs a name. The name is the most important means of communicating the pattern. Have a second look at the aliases you collected earlier. A good name makes it easy to remember the essential core of the pattern's solution. In most cases, noun phrases are best suited as catchy names. If, however, the pattern describes rather a process than a thing, a verb phrase might be more appropriate. In any case, the name should be short and easy to remember.

What name helps us remember the solution?

There is no moment in pattern writing that is optimal to think about the name. It is the solution distilled even further. That is why it will most probably jump to your mind when trying to catch the solution. But as with all other sections, it is hardly done after a single pass anyway.

Name:

DOOR LOCK

Context, Problem and Forces again

You have now set up all pattern sections required for a full pattern. This is the time to re-visit your problem domain.

Describing and extending the solution and its consequences seldom is difficult because you know the solution well. Distinguishing between Context, Problem and Forces often is more challenging.

In case you are not sure into which section a specific statement belongs, the following rule of thumb might help:

- Keep the Problem statement short and crisp.
- Everything you need to explain as background becomes part of the Context.
- Everything that describes why the Problem is difficult becomes part of the Forces.

Section Order

There are different ways to structure and label a pattern. The so-called 'Alexandrian Style' does not have any explicit sections at all [for an example see [2]]. The pattern format presented here has the benefit of being rather clear and simple.

Now, order your sections into the proposed pattern format: Name, Context, Problem, Forces, Solution, Consequences.

Congratulations: You just have finished your first pattern!

Let's see how the DOOR LOCK pattern has evolved by now:

DOOR LOCK



Context:

You are the owner of a house in an area where you cannot leave your property freely accessible to all people because this might attract thieves. The building can be accessed through doors.

Problem: You want to control access to the building.

Forces:

- **Selective access rights:** You want to ensure that only you and some selected other people have access to the building. Others shall not.
- **Protection:** You want to protect your property but you cannot be at home all the time or ask somebody else to watch your belongings. Even when you are at home you might want to protect yourself together with your belongings to have a peaceful sleep.
- **Effort of access:** Access to your property should be easy for you any time whereas it should be very difficult for any intruders to get inside.
- **Storage of permission:** Permission should be easy to use and reliable to store.
- **Change of permission:** While it should be easy to enable other people to get in, it should be easily possible to revoke this ability from them.

Solution:

Integrate a mechanism to all relevant doors of the building that can be repeatedly operated from both sides of the door by a matching key to allow or prevent the door from being opened.

Consequences:

Benefits:

- **Protection:** You can easily lock the door when you are inside your property and when you are leaving. It is hard for someone without the key to enter.

- **Effort of access:** A key is a lightweight object that can be stored and secured easily.
- **Storage of permission:** As the key is a physical object, you can easily take it with you any time so that you can always lock and unlock the door as desired.
- **Change of permission:** You can pass keys on to selected people to give them the opportunity to get into the building and you can ask them to return the keys any time.

Liabilities:

- **Selective access rights:** You need to physically distribute the keys to the persons who should get permission.
- **Storage of permission:** Because the keys must be difficult to duplicate, you typically only have a limited number of them. So you cannot give access to your property to as many people in parallel as you may like.
- **Change of permission:** Withdrawing a key from a single person is ok, giving a key to a single person as well. But if you fear that a key that you do not have access to any more is abused (or lost), you must replace the whole lock and all keys. Otherwise, somebody who finds the key might easily get inside. Exchanging lock and keys can mean considerable effort.

Examples:

A locked front door effectively keeps people from entering your home. It needs criminal energy to get across it. If go on holiday you can easily give the key to somebody you trust to water your plants and feed your pets.

A garage door can be locked to protect your car from being damaged or stolen. However, if you lend your car to somebody else while it is locked into the garage make sure you give him your car key AND the garage key. One alone is pretty useless.

Related patterns:

A PADLOCK shares many characteristics but it cannot be operated from both sides of the door. As an advantage it can be changed easily.

A COMBINATION LOCK can be opened by everybody who knows the access code, which can be easily communicated without passing physical objects. You cannot know, however, how many people know the access code and have passed it on.

As we did in the DOOR LOCK pattern, we would like to encourage you to extend your pattern by an example and a picture/diagram. You can also think about related patterns, sources etc. and add them when applicable.

Next steps

Put your pattern aside for a while. Once you get back to it, you will be surprised how many aspects you can improve. As the authors did for the DOOR LOCK pattern, it takes several iterations of refinement to distill the actual pattern into a readable and easily understandable form.

How do you know whether your pattern is good?

First of all, make sure that your pattern is complete. That is, make sure it contains all sections mentioned above and that these sections answer the questions that constitute them. Then, pass your paper on to colleagues and friends and gather as much feedback as possible. They can tell you how valuable your paper is to them.

Even old hands find it very helpful to get a second opinion on their patterns. To facilitate this, the pattern community has established a mechanism called 'shepherding'.

Your first pattern does not need to be perfect!

Shepherding is a process in which a pattern author receives feedback from another, experienced pattern author. Shepherding is an iterative process. Typically, the author incorporates the feedback he or she received and sends a revised version to the shepherd who in turn gives another set of comments. This form of personal coaching helps all authors, regardless of their experience, enormously to get ahead.

Shepherding is prerequisite for all papers that are accepted to a pattern conference of the PLoP series. So, if you would like to benefit from having a shepherd, you just need to submit your paper to a PLoP conference ([5], [6]). These conferences take place mainly to facilitate giving and getting feedback on pattern publications. The shepherded paper is discussed at a writers' workshop at the conference in a very rewarding style.

How do you find a shepherd?

Submit a pattern paper and find out yourself!

Further Reading

There is much more to be said about patterns in general and how to write them. The best follow-up would be reading *Advanced Pattern Writing* ([1]) as this paper proposes an approach to pattern writing that is very similar to the approach described here. Yet, the pattern language digs a little deeper. It builds on *A Pattern Language for Pattern Writing* ([2]) that presents even more patterns how to write a pattern. It extends the scope from single patterns to pattern languages. Finally, there is *The Language of Shepherding* ([3]) that is actually targeted towards shepherds but also contains a lot of information for writers.

A comprehensive general introduction to the world of patterns can be found in *POSA 5: On Patterns and Pattern Languages* ([4]).

Acknowledgments

First, we would like to thank Didi Schütz. Several discussions with him helped us gain deeper insight into the art of pattern writing ourselves. This paper was submitted to EuroPLoP 2011. We are very thankful to our shepherd Neil Harrison who pointed out many important aspects how to improve the paper before and even after the conference. During the conference, we got many further suggestions for improvement from the participants of workshop C: Klaus Marquardt, Markus Gärtner, Mandy Chessell, Jörg Pechau.

References

- [1] Neil B. Harrison: Advanced Pattern Writing – Patterns for Experienced Pattern Authors, in: Pattern Languages of Program Design 5, edited by Dragos Manolescu, Markus Voelter, James Noble. Addison-Wesley, 2006.
http://hillside.net/europlop/HillsideEurope/Papers/EuroPLoP2003/2003_Harrison_AdvancedPatternWriting.pdf

- [2] Gerhard Meszaros, Jim Doble: A Pattern Language for Pattern Writing, <http://hillside.net/index.php/a-pattern-language-for-pattern-writing>
- [3] Neil B. Harrison: The Language of Shepherding, <http://homepages.mcs.vuw.ac.nz/~kplop/Shp.html>
- [4] Frank Buschmann, Kevlin Henney, Douglas C. Schmidt: Pattern-Oriented Software Architecture 5 – On Patterns and Pattern Languages. Wiley, 2007.
- [5] EuroPLoP conference website: <http://www.europlop.net>
- [6] PLoP conference website: <http://www.hillside.net/plop>