

The structure of patterns

Christian Kohls
c.kohls@iwm-kmrc.de

Third draft – September 18th 2010

Abstract

This paper aims at learning more about the structure of patterns by examining a very simple and intuitive example of a pattern: a path as a solution to reach a goal. A path is obviously bound to its environment and this happens to be a good starting point to emphasize why the context of a pattern is so important. By visual illustration one will find that it is the environment with its embedded forces that shapes the possible solutions. On a map this relation can literally be seen. Paths and their representations on maps are very well known concepts. This allows the use of common sense to understand properties of patterns that are otherwise hard to grasp. The difference between patterns of things in the world and their description as patterns becomes clearer. Fuzzy terms such as “medium level of abstraction” or “wholeness” become meaningful if you consider them in a context where they are straightforward.

Motivation

Design patterns are well established in the domain of software. Other disciplines, such as human-computer-interaction (Borchers, 2001; Tidwell, 2005; Schümmer & Lukosch, 2007), web design (Van Duynie, Landay, & Hong, 2004; Mahemoff, 2006; Scott & Neil, 2008; Malone & Crumlish, 2009), management and organization (Harrison & Coplien, 2005; Rising, & Manns, 2005) or pedagogy (Anthony, 1996; Bergin, 2001; Derntl, 2006), have adopted the pattern concept.

Since all disciplines strive for the discovery of general rules, laws or patterns it is sometimes hard to explain what is unique in the approach taken by the pattern community and the ideas based on Christopher Alexander. Terms such as “context”, “forces”, “problem”, “consequences”, “diagram”, or “interaction” have particular meaning and dependencies in this approach. I have found it helpful to use proven paths in a territory to explain how patterns capture, describe and explain good practices. A path is an example of a solution that leads to a specified goal and is a simple concept that is very well known to everyone. This simplicity enables us to show how the territory and the overall situation define which paths are potential solutions and which properties each path has.

Pattern sceptics very often do not understand what is innovative about the pattern format. Best practices have been documented for a long time and scientific discovery is all about finding invariants – patterns – in nature. While that is true, the elaboration in the pattern format does not only describe recurrent forms but reasons about the form in terms of forces and consequences. Most people agree that a solution has to be adequate for the task at hand. However, many descriptions that do not follow one of the pattern formats lack this context information. The purpose of this paper is to use a very trivial metaphor to address common sense. Nobody would follow directions that belong to a different landscape. Nobody is happy about directions that are too abstract (e.g. missing one or two turns) or too detailed (e.g. describing every stone of a path). Nobody would follow directions unless they make sense, i.e. it should be reasoned why one should follow a path. Yet many descriptions of solutions do not provide this information. They are too abstract or detailed and fail to explain why a form solves a given problem. Patterns, on the other hand, do capture contexts, forces, consequences and solutions on a medium level of abstraction.

Pattern beginners who are willing to share their knowledge by writing patterns will see that it takes more than just knowing a proven path to describe a reusable solution. Once we are aware of something that has worked several times we are eager to write it down. But it is much harder to write down why something has worked and to reason under which other circumstances the configuration will work or fail. The path metaphor hopefully shows that it is worth the effort. By explaining a solution form along with the fitting environment and what has shaped the solution we can help the user to understand the form and make sense of it. In order to adapt a pattern to a specific situation one should know what has influenced the general solution.

Pattern experts can use the visualizations to discuss their conceptions about patterns. Both agreement and disagreement will give new insights to how we see patterns. Since human beings are doing very well in perceiving visual patterns, some properties of recurrent structures might be grasped more intuitively, e.g. how patterns can be divided into sub pattern, how patterns overlap and how patterns work together. Last but not least, the path metaphor might help to teach the core concepts to interested novices or to convince sceptics of values the pattern format offers.

A basic definition

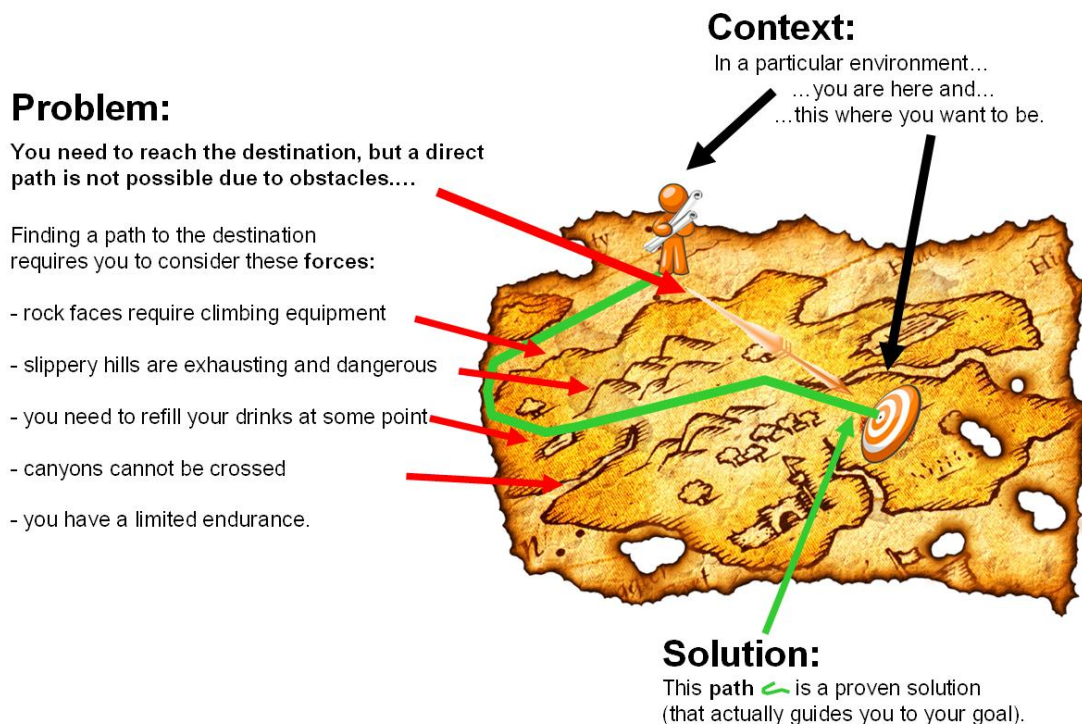
Let us begin with Christopher Alexander's definition of patterns in *A Pattern Language* (1977):

„Each pattern describes a **problem** *which 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*.“ (Alexander, 1977)

There are two statements in this definition that are essential to patterns:

1. a pattern is the solution to a problem in an environment
2. the form of the solution can be used over and over again without ever doing it the same way twice

Both statements, simple as they may appear, are rich in their meaning. Let us therefore consider a familiar environment, a hillside landscape, to explore the concept of patterns. In such an environment, if we intend to hike from one place to another, we may encounter the problem that we cannot directly move in a straight line to the destination because we cannot cross canyons or pass over steep walls. Each trail that guides us to the destination is therefore a solution to our problem: getting us from where we are to our desired destination.



Understanding context

A context is the environment or the situation we find ourselves in. In the simple example of a hiking tour, the context is the physical environment of the landscape as well as your current situation (that is where you are, how well trained you are, what your hiking skills are, which tools you have at hand etc.) and what your goal or intent is. Besides reaching your final destination, your intention may be to find ways that are not too tiring or to find ways that are physically challenging. You may desire that the path should lead along scenic views or cultural heritages and make this a requirement for the solution.

The context depends to some extent on your personal intent and preferences while at the same time the environment – the hills, canyons, slopes, lakes, cabins - is out of your hands.

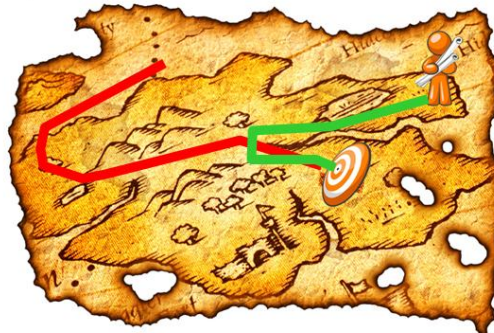
At first glance, the context of that landscape seems to be fully fixed and static. However, the context can vary. One can hike at different times of the year, there will be different animals that cross the path, and there may be different groups of people who take the same trail. Yet what matters in the context - the shape of the landscape - remains invariant.

Also, we have to understand that the form of the solution is only adequate and of value if it fits to the context. For example, the very same form is no longer a solution if an essential part of the context (such as our starting position) changes.

The solution pattern  only fits to a certain context.



If the **context** changes (for example we start at a different position or we are in a different environment) the solution pattern changes.



The **core problem** that we cannot pass all areas remains but the specific **forces** change.

A change in context can make new paths possible as well (e.g. a bridge over a canyon) or deny paths that have worked in the past (e.g. a rock slide blocking the way). The context implies which solutions could work. In a changed context the same solution form may not fit any more. Another context sets new constraints, boundaries, introduces new requirements and offers different opportunities. Constraints, boundaries, requirements and opportunities are different types of forces. Forces can support each other or raise a conflict if they imply different things to do. Such conflicts are problems that need to be resolved. Problem and forces of a given context are usually discussed in separate sections because they are important to understand why a pattern is shaped the way it is.

Understanding problem and forces

If forces are the reason to build a solution in a specific way that works, we can consider forces as the cause that leads to the pattern. For example, what is the cause for the problem that we cannot go straight to our destination? Originally, it is the rock face that blocks our way. It **forces** us to take another way. That particular force is the original **problem**, the one that is problematic in the situation we are in. If we are looking for a way that brings us to our destination we have to account for **all the forces** that influence our chosen path in the given context. There are some forces that cause the original problem and there are more forces that shape the path of the solution.

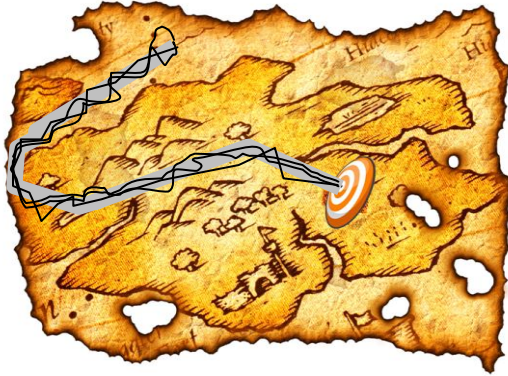


The red arrows denote forces that deny us a specific path, e.g. we cannot cross over a canyon (2) or climb a steep wall (1). The green forces, on the other hand, are attractive ways to go. There are also some attractive places to which green forces point, such as a place to get fresh water (3), or the actual destination (4).

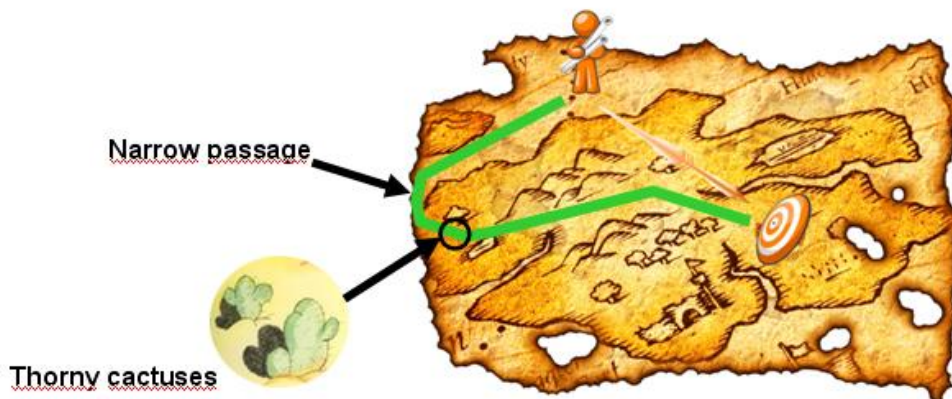
A solution is one known way that takes into account all the forces that matter in a specific context and addresses them to a satisfying extent. The form of the solution is a path that has proven in the past to actually lead to the intended goal and takes the forces appropriately into account.

Understanding solutions

The term **pattern** suggests recurrent solution forms. And that is actually what happens if people use a trail: no two hikes along a trail will be identical. We will use detours, walk crisscross and unfold the hike every time in new a way while still following the same way. That refers to Alexander's second element of the definition of patterns. It is a solution *which occurs over and over ... without ever doing it the same way twice.*



While there is variance, the general structure is preserved. Every concrete move along a path has the same structural quality even if each instance unfolds differently. The range of possible variation is different for the various parts of a path. At some points the space of a path may be broad, allowing many different steps to move along the same path. At other points, the path may be narrowed or obstacles only allow specific moves. Those points are critical and need more attention to make the path a success.



Likewise, for implementations of software patterns there are parts that can be altered to different extends. An OBSERVER (Gamma et al., 1995) always has an update() method. But a concrete implementation of an OBSERVER may have several update-methods, each having a different name. This could be a fast-forward way to update different things in the observing objects. Another option would be to provide a parameter to the update() method that defines what should be updated. This parameter could be a simple flag or an object that implements a STRATEGY for the update. These are three different implementations but each would be coherent OBSERVER behaviour. What works best depends on the concrete context. The range

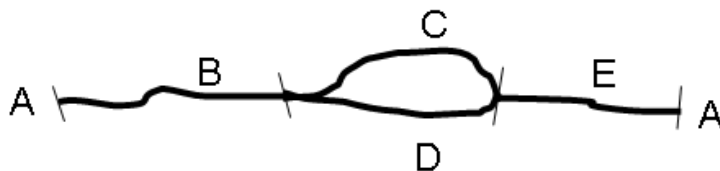
of method names and parameters potentially provided is very broad. The interactions and responsibilities, however, do not vary very much.

An example for a pedagogical pattern is an ONLINE TRAINING (Kohls, 2009). An ONLINE TRAINING typically consists of examples and is presented using a web conference system. There is a wide range how many examples and what types of examples are used during trainings. But there are only a few conference systems (narrow range) that are adequate for online trainings.

The pattern ONLINE TRAINING also lists a number of obstacles. Once we have made our decision for one path we may encounter obstacles on our way. The path may lead to our goal but we have to take care of some challenges. These challenges are local problems that occur in the context of the chosen solution. It is common to describe recurrent local problems and their solutions as separate patterns that can be used in the context of the current pattern. That is the reason why solutions frequently refer to other patterns.

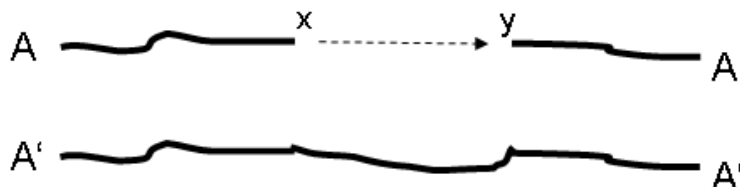
Noble (1998) distinguishes between the primary relationships that “a pattern uses another pattern, a pattern refines another pattern, or a pattern conflicts with another pattern”.

A pattern that uses another pattern is a meta-pattern and describes a solution that relies on sub solutions. For example, the solution of an ONLINE TRAINING suggests that there are PREPARED EXAMPLES and PAUSES FOR QUESTION. Both are sub-patterns that make sense in the context of ONLINE TRAININGS. Paths, too, can be composed of sub-paths. One could say “Go to the Factory. Then go to the bridge. Finally you reach the castle.” This may describe the path from a current position to a castle in general terms. Each of the directions is a sub-path that can be described on its own if necessary. Someone familiar with the territory may find the way to the factory without further help whereas other persons need directions how to get to the factory. Then “Go to the Factory” should be a separate description of a path.



Example: path A uses paths B, either C or D, and finally E. There is a pattern sequence because the order in which the paths are followed is apparently important (one cannot use E directly after B). At which points a larger pattern can be divided into smaller patterns is discussed in the section “Wholeness” because not every division is meaningful.

In the previous example, the meta-path A does not prescribe whether C or D should be followed. Which one is better depends on the context. Maybe D is preferable on a rainy day. Both sub-paths C and D are in the range of possibilities to follow path A. Pattern refinements are more specific about the actual solutions because they discuss the context and solution on a less abstract level. For example, if we include weather conditions in the context, then on a rainy day D is imperative and we can be more specific about the form of A' which refines A.



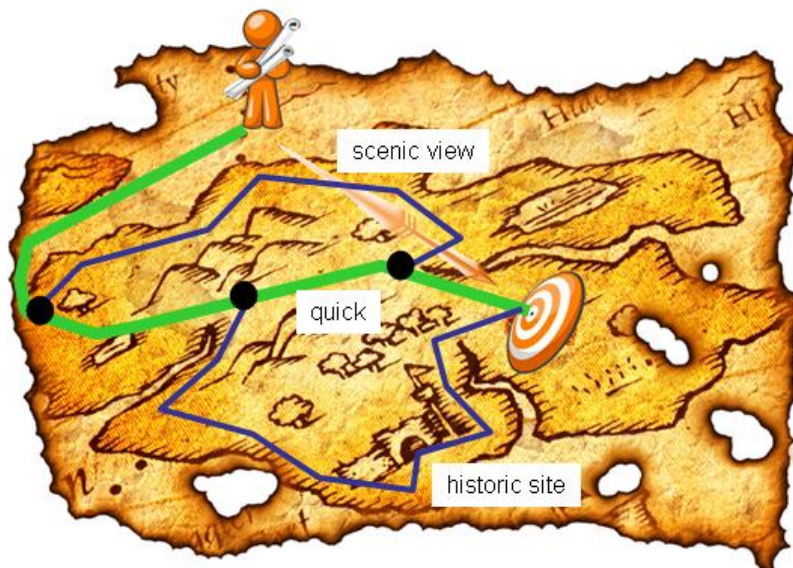
While A leaves it open how to get from x to y (either following C or D), A' provides an answer because it matters in the more specific context. Finding an appropriate level of abstraction for a target audience is not trivial. You will find more considerations in the section “Abstraction”

Dividing patterns into sub-patterns is not the only way to reduce the complexity and allow new combinations. This approach is additive – fitting patterns are used together to form a whole. But patterns can also be used in a multiplicative way. That is, using overlapping patterns cooperatively at the same time. If one follows a pattern of steps, other patterns such as RAIN COAT, BACKPACK or WATER BOTTLE are supporting that pattern at the same time. More precisely, they are modifying the pattern: it is different to walk along a path with our without a BACKPACK. Kolfshoten & Santanen (2007) describe this general principle as Modifiers.

However, it is not possible to move along two physical paths at the same time. If there are two alternatives that lead to the same (sub) goal one has to decide for one. The paths C and D are such competing patterns because they both have the same starting and end point. Which path is favourable depends on the specific context. To help designers decide which one to choose, pattern descriptions should capture the consequences of each of the alternative solutions.

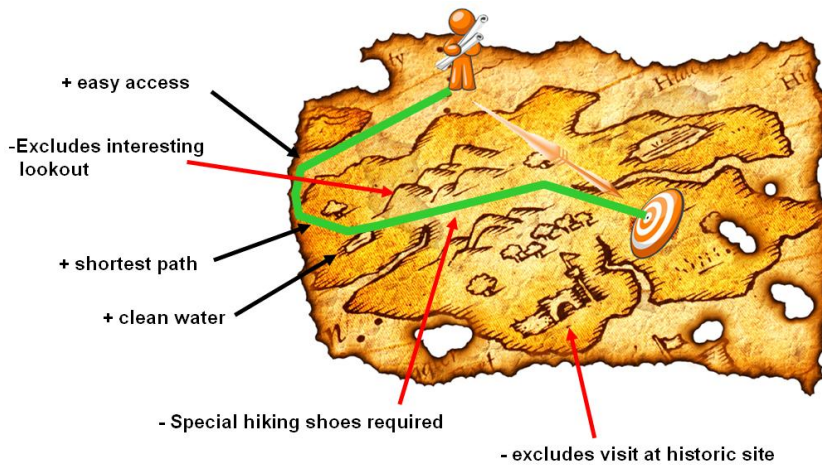
Understanding Consequences

The context shown on the map allows more than one solution path. For example, the path that leads to the historic site certainly has the advantage of visiting that site. However, it is a way that requires more time.



The two paths compete with each other. In the process of finding a decision the pros and cons are weighed against each other; the consequences are compared. The overall situation will determine the actual path taken. If the hiker has a special interest in historic sites this interest “forces” him to take the longer way because he “has” to see it. In this context the path that runs along the historic site is best for that hiker. However, in a slightly different context there might be a time constraint to reach the goal. In that case the hiker might be forced to take the shorter path because reaching the goal in time (e.g. before sunset) might be more important than seeing the historic site.

The better we understand the consequences the better we can find an appropriate decision. If we know the advantages (values) and disadvantages (costs) of a path we can decide better which one fits the problem at hand.

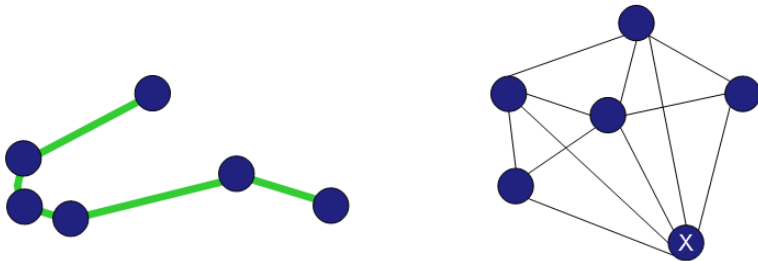


Which consequences matter is directly related to the specific situation. Small changes in the context (time constraints, personal interests) have impact on the path we consider best. There are also situations in which two paths are equally valuable. For example, if one path leads to a historic site and the other to a scenic view, both being equally interesting for the hiker. In that case the decisions might depend on mood or previous experience (if the hiker has seen already five historic sites he might choose a scenic view). Both are contextual properties as well. Unless you flip a coin each decision is based on the fact that one path is favoured over the other. It is also likely that there are paths that fit better to the context, but that have not been found yet (e.g. a path that allows the hiker to reach the historic site and be at the goal in time).

The choice of a path has also consequences for the next steps to do. Once it is decided where to go there is a resulting context that suggests which sub-paths can be followed and which preceding and succeeding paths are available. The start and end point of a path have to connect to surrounding path ways. The steps that can follow depend on previous decisions.

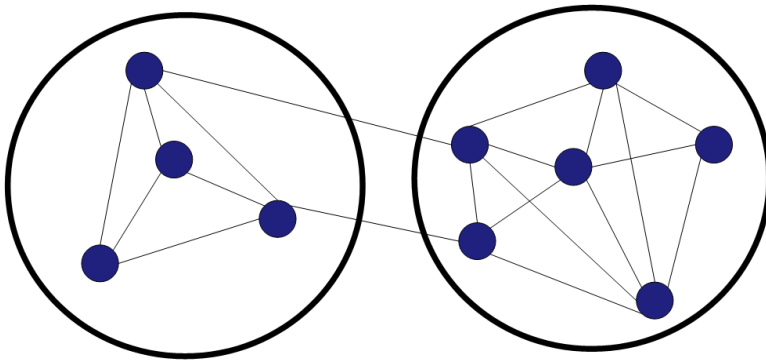
Wholeness

On a simple path the next step only depends on our current position and environment. However, most solutions are more complex. For example, if a hiker has followed a path that allowed him to fill up his water bottle and saved him energy, then he is more flexible to choose between alternatives.



In the right picture we see a form in which the elements have more than one relation to other elements. The element marked X has five connections. Thus, any change of X will have direct impact on five other elements.

The inner dependencies make the solution a complex system. However, not all design decisions depend on each other in the same way. Some variables are only related to a few others. For that reason we can split the design into sub-problems that can be tackled independently to a larger extent.



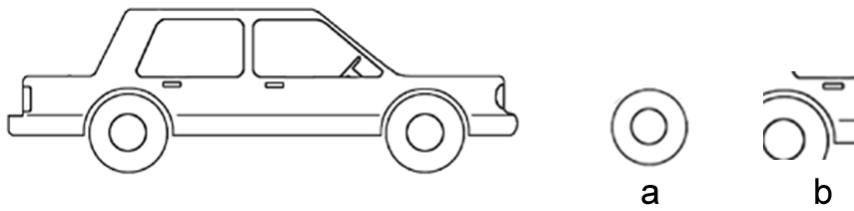
(Based on Alexander, 1964)

Such a part can be formed to meet the forces of the context almost independently from the rest of the solution. Only where two parts of the solution meet (or two parts of a path connect) there is a dependency. If there are three path sections A, B, and C, then B has to fit to A and C because they are now part of the context for B.

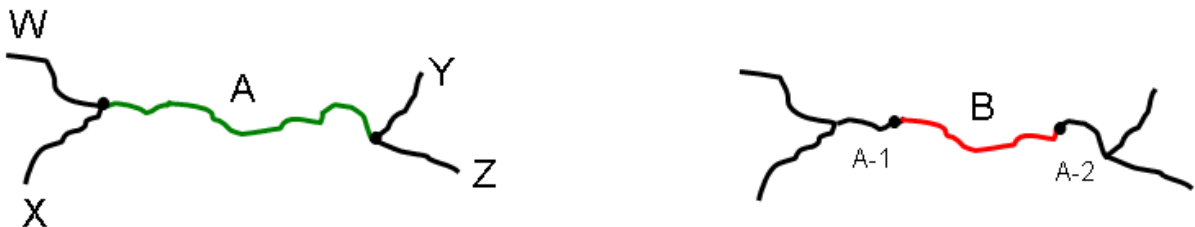


Left: B fits to A and C. Right: B does not fit to C. The parts are no longer a solution.

Yet it does not make sense to split a solution at arbitrary points even if the parts fit together. For example there are different ways to split a car into parts:



While both parts recur with the same frequency in the world, only one is perceived as a whole part. The wheel (a) is self-contained and whole. Likewise, splitting a path into parts only makes sense at some points.

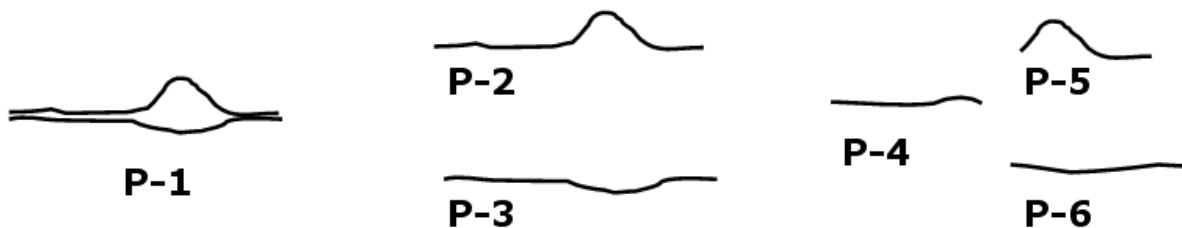


Sub-path A is whole because it covers the whole section between two decision points. There are different ways to get to the beginning of A and continue afterwards. Sub-path B is not whole as it starts and ends “in the middle of nowhere”. It is not self-contained because it depends on the extension A-1 and A-2 which are implicitly part of A. The whole section of A belongs together and any sub-part of A is not whole for it cannot stand on its own.

While B can only be used in one context, A can be used in multiple contexts. A hiker can use A if he comes from W or X and continue to Y or Z.

The path A is open for combination into new paths: W-A-Y, W-A-Z, X-A-Y, and X-A-Z. Path B allows only one combination: A1-B-A2.

However, there are also alternative options to split up a path into whole sub-parts. Let us consider two simple paths to illuminate the challenge:



To describe the two paths (or patterns) we could either

- Describe P-1 including the variation in the description
- Describe P-2 and P-3 as separate descriptions
- Describe P-4, P-5 and P-6 which can be combined to build each of the structures

As a heuristic, how we divide the paths (or patterns) depends on the amount of similarities and differences.



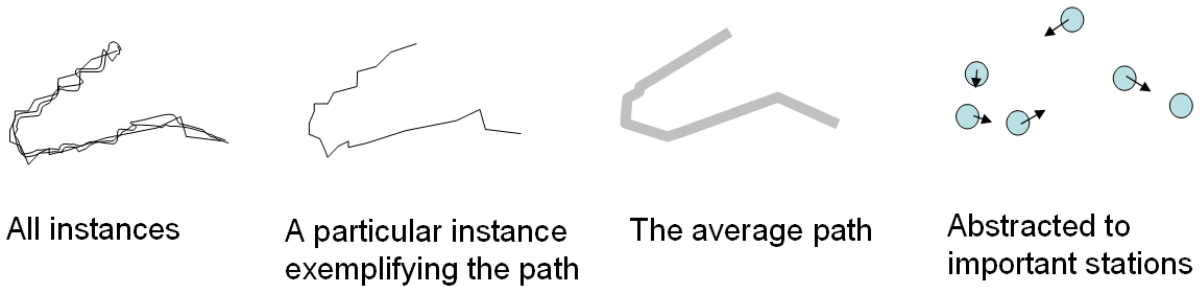
For example, two structures as in A would be captured in a singular description that contains the variation. Two structures as in B would be best described in two separate description that are redundant in only some parts. In C we see two paths that split up at a certain point. One could describe the common path in a single document and the two alternatives in two additional documents.

Abstraction

The previous sections have discussed how a specific path solves a problem if it fits to the context and given forces. A map has illustrated how the conditions of the territory offer different paths and also set boundaries. Such a map can clarify the structure of the path and the environment. But it must not be confused with the real territory and the actual paths.

In the same way pattern descriptions must not be confused with the actual patterns and solutions in the world. This is often overseen when we refer to “a pattern”. Sometimes “a pattern” means the actual recurrent structure in the world, or at least what an individual perceives as recurrent and invariant (Kohls & Panke, 2009). In other cases “a pattern” refers to a description of a pattern which is usually captured in a typical format (often referred to as “pattern format” or “pattern template”). Gabriel (2002, p. 175) writes: “A pattern, then, is both something in the world – the configuration found in excellent artefacts – and a literary form, that is, the written description of the physical configuration and why it should be built.”

The pattern description is not the pattern itself but it incorporates different perspectives to represent the pattern. To illustrate this, consider the different options to represent the various instances of a path. One could use a single instance that leads along the path by example. One could also use the average positions of all instances in which the trail was walked on. That would be an abstracted representation. We could even further abstract by only showing important hallmarks that have to be crossed and leave open how these milestones are reached:



Like a map, a pattern description has to provide enough information to actually follow the solution path. It will leave out details and concentrate on hallmarks that help designers to find their way. But it must not be too abstract. Otherwise we are losing the form and the specific structure that guides the designer through the territory.

Mapping is the art of representing the relations of the real world in a diagram. Geographic maps (as used in the examples) represent the spatial relations of a territory. UML class diagrams are mappings of hierarchies. Instead of using formal maps, informal maps tell the story in a different way. Classic geographic maps have a bird's eye perspective. However, when you ask for directions you often get instructions where to go next ("turn right at the third intersection"). Such directions are sometimes supported by demotic maps that are scribbled and look like drawings on the back of a napkin. Doebely-Honegger & Notari (2010) have used informal didactic process maps to let teachers illustrate what happens in their lessons. Such diagrams are easy to create and understand but still show the important relations between the elements. The point is that informal ways of describing forms (such as paths or patterns) are very often quite precise because they take into account what actually matters in the given context. Sometimes a stone may be ignored, at other times the stone can mark a change of directions, and sometimes it is a spot of danger. Whether or not the stone is included in the map is not governed by formal rules but depends on its importance finding the right way.

Similarly, a pattern description should include those parts of the context that are critical to understand when and why a solution works and what has shaped the solution. Important consequences (e.g. the advantages, disadvantages, liabilities or further directions) should be highlighted and explained.

Conclusions

The path metaphor hopefully illustrates appropriately some important concepts of patterns. Actually the path is more than a metaphor. It is an example for a pattern because each path solves a problem of getting safely from one point to another. In our everyday language we speak of **ways** to achieve something or **step-by-step** solutions. The original meaning of **method** is "a way to a specific goal". If a path is an example for a pattern then all properties of patterns should apply for paths as well. Hence, we can use the visual nature and common familiarity with paths to explain the properties to beginners and sceptics.

I have focussed on explaining context, problem, forces consequences and solution using the path example. Yet all pattern formats I know use more description sections. The reason is that we can elaborate each of these views in more detail. For example, very often the **intent** is explicitly stated in a description field. The intent is part of the context or situation. The forces, too, can be described in several sub sections. Some patterns have a **rationale** section that

explicitly reports how the forces have been resolved by the solution. The solution itself is tackled from multiple perspectives, including **summaries, diagrams, discussions, examples** or **stories**. The **consequences** are often sub-divided into **advantages, disadvantages, liabilities** etc. Which format is chosen depends on both the preferences of the authors and the domain of the pattern. If a hiking path is described it might be a good idea to include seasonal conditions as a section in the context statement. Educational patterns sometimes state the group size or duration of a method explicitly. There is not one golden pattern format. However, the context, problem, forces, the solution and its consequences should be captured in each pattern. Also, patterns should always have an expressive name to make it part of the vocabulary of a designer.

Another section often found is **related patterns**. While patterns are self-contained they do not exist in isolation. Patterns can work together or set the context for other patterns. That is why in a context section there are many references to other patterns by using their names in capitalized letters. Likewise, a pattern that sets the stage for other patterns refers to those patterns in its solution part. This paper has discussed some of the relations. There are limitations of the path example that do not show the complexity and flexibility that is found in other patterns. For one, a path only has to fit to its local environment and to the previous and succeeding sub-path. But there are patterns that have to fit to a larger number of other parts at the same time. Another special property of physical paths is that a person can only choose one path at a time. Large design projects on the other hand usually require that users follow multiple (metaphorical) paths at the same time. In software architectures, the same class can play a role of multiple patterns at the same time. This raises the question which types of patterns can be combined with each other. The pattern of a path can be combined with other patterns such as WATER BOTTLE or RAIN JACKET simultaneously. But the combination with other physical paths can only take place in a sequence. Is that because paths are patterns of the same kind whereas utilities such as water bottles and clothes are of a different kind? This question is beyond the scope of this paper. But it shows that the path example is not only a tool to introduce patterns but stimulates new questions

Acknowledgment

The author would like to thank his shepherd Rebecca Wirfs-Brock. Rebecca has encouraged me on the path of writing this paper and has helped me to overcome obstacles and avoid dead-ends. When I was running into the wrong directions, she has shown me valuable ways to improve the paper.

References

- Alexander, C. (1964). Notes on the synthesis of form. Cambridge: Harvard University Press.
- Alexander, C., Ishikawa, S., and Silverstein, M. (1977). A pattern language: towns, buildings, construction. New York: Oxford University Press.
- Anthony, D.L. (1996). Patterns for Classroom Education. In Vlissides, J., Coplien, C.O., & Kerth, N.L. (Eds.). Pattern Languages of Program Design 2. p. 391-406. Reading, Mass: Addison-Wesley.
- Bergin, J. (2001): Fourteen Pedagogical Patterns. Proceedings of the 5th European Conference on Pattern Languages of Programs. pp.1-40. Konstanz: Universitätsverlag Konstanz.
- Borchers, J. (2001). A pattern approach to interaction design. Wiley series in software design patterns. Chichester, England: Wiley.

- Derntl, M. (2006). Patterns for person centered e-learning. Berlin: AKA.
- Doebely-Honegger, B. & Notari, M. (in press). Visualizing learning processes using Didactic Process Maps. In Kohls, C. & Wedekind, J. (Eds). Investigations of E-Learning Patterns: Context Factors, Problems and Solutions. Hershey: Information Science Pub.
- Gabriel, R. P. (2002). Writers workshops and the work of making things: Patterns, poetry. Boston [Mass.]: Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley.
- Harrison, N., & Coplien, J. O. (2005). Organizational patterns of agile software development. Upper Saddle River, NJ: Pearson Prentice Hall.
- Kolfschoten, G.L., & Santanen, E.L. (2007). Reconceptualizing Generate thinkLets: the Role of the Modifier. HICSS 2007: 16
- Kohls, C. (2009). A Pattern Language for Online Trainings. Proceedings of the 14th European Conference on Pattern Languages of Programs.
- Kohls, C., & Panke, S. (2009). Is that true? Thoughts on the epistemology of patterns. Proceedings of the 16th Conference on Pattern Languages of Programs. Chicago.
- Mahemoff, M. (2006). Ajax design patterns. Sebastopol, CA: O'Reilly.
- Malone, E., & Crumlish, C. (2009). Designing social interfaces. Sebastopol, CA: O'Reilly Media.
- Noble, J. (1998). Classifying relationships between object-oriented design patters. Australian Software
- Rising, L., & Manns, M. L. (2005). In Fearless change: Patterns for introducing new ideas. Boston: Addison-Wesley.
- Schümmer, T., & Lukosch, S. (2007). Patterns for computer-mediated interaction. Wiley series in software design patterns. Chichester, England: John Wiley & Sons.
- Tidwell, J. (2005). Designing Interfaces. Sebastopol: O'Reilly Media.
- Van Duynie, D., Landay, J. A., & Hong, J. I. (2004). The Design of Sites. Boston: Addison-Wesley.