Is that true…?

Thoughts on the epistemology of patterns

Christian Kohls
Knowledge Media Research Center
Tuebingen, Germany
c.kohls@iwm-kmrc.de

Stefanie Panke
University of Bielefeld
Bielefeld, Germany
s.panke@iwm-kmrc.de

There is nothing more practical than a good theory.
Kurt Lewin

In theory, theory is the same as practice, but not in practice.
Fnord Bjørnberger

This paper presents a theoretical perspective on patterns derived from epistemology and theory of science. We argue that patterns are specific kinds of theories and that the process of pattern mining is similar to scientific discovery. Exploring the concepts induction, deduction and abduction with respect to patterns, we reflect upon common methods of pattern mining in the pattern community. This allows for a critical discussion of the level of confidence and corroboration of patterns. We suggest new research questions on the mining and evaluation of patterns.

For the scientific scholar the paper offers arguments that pattern mining is a research process with outcomes as reliable and sound as other procedures. This justification is needed to establish the pattern approach as a scientific methodology beyond the scope of the pattern community. At present, it is unusual to present scientific findings in the pattern format. Usually, patterns are just illustrative examples in research publications. We will argue that patterns are theories and cannot be separated from theoretical findings.

For the pragmatic oriented pattern practitioner, e.g. users and authors of patterns, this paper encourages the critical reflection of the pattern concept. Patterns are not tried-and-true per se, just like theories they have to be objected to empirical tests. Discussing the validity of patterns is not just belle étage philosophy. Understanding the epistemological nature of patterns is crucial to derive criteria for pattern quality, e.g. the degrees of corroboration, and the limits of objectivism – especially since patterns are not only descriptive documentation but normative instructions, designed to have an impact on shaping our environments.

PLoP 2009 Submission Writer's Workshop version - 2009 August 13th

1. Introduction

Whether or not patterns can be called scientific methods has filled the beer cellar of Kloster Irsee with heated discussions among “practitioners” and “researchers” during many EuroPloP conferences. The practitioners usually reject a scientific approach to patterns, arguing that good patterns contain “nothing new”, but capture existing knowledge. From their point of view, the nature of patterns is a specific and very useful genre for technical documentation. Unsurprisingly, pattern researchers beg to differ. They consider pattern mining as a scientific endeavor. Patterns reveal previously unreported regularities. In this paper, we try to reconcile both views, by distinguishing patterns that represent scientific progress from patterns that are just another – albeit effective – genre for documentation.

If patterns truly contain “nothing new”, the question is: What is the point of stating the obvious? There are good reasons to consider the documentation of recurring regularities in different design projects as scientific progress. The argument that there is “nothing new” in a pattern must be rejected; otherwise there would be nothing new to physics either, since physical objects and the laws of physics have been around before, just as design objects or programming styles have been around before somebody sets up a pattern language or collection. The discovery and description of a new species is without question considered scientific progress. Of course, the animals are not new – they lived there before – but they are newly discovered.

What is the difference? Why do people perceive what biologists and physics do as “discoveries” and the work of the pattern community as “documentation”? The answer lies in the nature of patterns: Writing a pattern is not “the same” as inventing the light bulb or discovering the motion of the earth around the sun by carefully and systematically staring through a telescope. Patterns describe the functional (problem resolving) properties and forms of objects, artifacts or social practices. In the process of pattern mining, we look at established ways of programming, manufacturing, producing, teaching, etc., and then specifically describe those elements that have a specific function in a given context. To capture these invariant features and to make the necessary generalizations is a creative process that generates knowledge. For instance, documenting various forms of Indian rain dances and then concluding that they provide social cohesion in the context of rough living conditions caused by a draught is unquestionable scientific progress.

However, not all pattern descriptions contain new findings. If we describe an elephant in the pattern format, this can hardly be called scientific progress – simply because the functional elements of an elephant’s physiognomy are well known facts in the biologists’ scientific community. In that case, the pattern writer delivers just another form of presentation. There can be many different descriptions of elephants, and each description can be easier or harder to understand, more or less entertaining. There is “nothing new” with regards to content. Nevertheless, the format of documentation itself can be innovative. Still, one can conduct research on such patterns: It is an interesting question whether knowledge represented in the pattern format is more accessible than for instance knowledge maps, advanced organizers, unstructured text, etc. Also, different pattern descriptions of the same patterns can be compared in empirical studies. In these cases, we are not interested in the content of the pattern, but in improving the pattern format and description.

Last but not least, research in other fields can also contribute to the content of patterns. As Buschmann, Henney and Schmidt (2007) point out, patterns and pattern descriptions evolve over time. Including new findings, e.g. new relevant forces, new consequences, new contexts or limitations, means to get a better understanding of the nature of a particular pattern. Therefore, pattern descriptions should be open towards inspiration by scientific progress, for instance the discovery of new materials, new procedures or new findings in human-computer-interaction.

While we can directly observe the single instances of design, e.g. a specific illustration, we cannot directly perceive the generalized forms and the causes for a form as they are captured in a pattern. Hence, patterns are purely speculative, and, comparable to a theoretical assumption, need empirical evidence. What we have learned from constructivism is that the apparent “objective reality” and the subjective meaning, value and volition are not a completely different kettle of fish. The observer is always part of the story and patterns are not neutral descriptions. Similar to the communication researcher Paul Watzlawick (1976), who posed the question: “How real is real?” we can ask: “How real are patterns?” Patterns have to be open to methodological falsification, since humans tend to find patterns everywhere – even in randomized circumstances. Consider the following two examples taken from Watzlawick (1976).

E1: In the late fifties, the city of Seattle was beset by the mysterious phenomenon that windshields on more and more cars were damaged by tiny scars, holes and pits. The affected citizens developed several theories including recent atomic tests by the Russians contaminating the atmosphere or erosion caused by a new local highway program. In the end, President Eisenhower sent a team of experts from the National bureau of standards to investigate the matter. These experts established the fact that there was no windshield picking at all. Instead, mass hysteria explained the apparent pattern. There was no epidemic of windshield pitting, but of windshield *viewing*.

E2: People are presented with a multi-armed bandit to participate in a problem solving experiment. Their instruction states that they have to press a certain numeric pattern to successfully activate a buzzer. What the subjects do not know is that the reward buzzer works totally non-contingent; nothing they do influences the buzzer. During the first part of the experiments, subjects receive a certain percentage of random rewards. During the second part, they receive no rewards whatsoever. In the third and last stage, they are rewarded every single time. At this point, all subjects are convinced they finally found the “pattern” of successfully operating the multi-armed bandit. Even after the experimenter tells the truth about the setup, most participants find it hard to believe, claiming they found a regularity the experimenter was unaware of.

As these examples show, our perception of pattern is wide open to delusion and what we find very much depends upon what we are looking for. It is therefore necessary to be able to distinguish between “correct” and “incorrect” patterns. This means that we have to investigate their content empirically and we also have to make sure that the structure of a pattern is logically sound.

2. Patterns are Theories

Discussing patterns as theories is not common in the community of practice that mines and documents patterns. On the contrary, there is a widespread consent on being suspicious and reserved towards theories and traditional science. Coplien (1998) stressed the point that patterns are made of “Stoff – real stuff, not platitudes and theories” and Rising (1998) highlights that “patterns are not theoretical constructs created in an ivory tower; they are artefacts that have been discovered in more than one existing system”. Another common ground in the community is the notion of patterns as being natural rather than artificial, comparable to an organic substance. Patterns are not “created artificially just to be patterns” (Buschmann et al., 2007). Similarly, DeLano (1998) observed: “Patters are not grown or created. They are present in the artefacts that already exist.” Besides being made of recurrent real stuff, patterns are considered to be true or proven: “good patterns are those solutions that are both recurrent and proven” (Buschmann et al., 2007). Patterns capture “well proven design experience” (Harrison, 1998), and they are “tried-and-true” (DeLano, 1998). Authors like Schümmer and Lukosch (2007) consider existing examples as proof of the patterns. Finally, there is nothing new or inventive about patterns, as they “are an aggressive disregard of originality” (Foote, 1997). Gabriel (2002) stated “software patterns are about describing what works and has worked well rather than finding new ideas”.

Yet, here we are, claiming that patterns should be regarded as theories, implying that their content is of hypothetical - not true, inherent or given - nature. Well, do not throw away the paper – we have our arguments.

2.1 A pattern is not the same as its manifesting artefacts

We are not opposing to the idea that the substance or base of patterns is (or should be) “real stuff” that has been observed or experienced. Our point is that there is an important difference between the object of a pattern and the objects that manifest the pattern. Just as there is a difference between the object of a theory and the objective facts which are explained or predicted by the theory. For example, Newton’s laws of motion are a theory and the actual motion behaviour is the “real stuff”. Of course, the laws of motion have always been at work and there is good reason to believe that people have been aware of them long before Newton was – otherwise the pyramids would not have been built (Berkun, 2007). However, Newton was the first who has systematically explicated the general law. In the same way, patterns do not invent or create the design solution they describe. Alexander (1979) stated that: “The task of finding, or discovering, such invariant field is immensely hard. It is at least as hard as anything in theoretical physics” (261). What is important for our pursuit is what we can learn for pattern mining from the methods of discovery, advancement, failure, falsification or proof applied in theoretical physics and other sciences.

Patterns are abstract entities that can only manifest in real instances. You can draw a pattern or represent it in other ways but what you are actually doing is to draw a diagram, sketch or model. It is an interesting question whether or not there is a “true” pattern behind the descriptions we give when we are writing pattern documentation. In that sense, the pattern is just an idea, in the similar way that specific triangles are not the same as the concept of a triangle. We cannot imagine the concepts themselves but only imagine them by exemplification – we cannot see “the triangle” but only “a triangle”, we cannot see “the beauty” but only “a beauty”. In the Platonic world view, all truth exist a-priori and sensual data offers only fuzzy representations of the original ideas. Hence, there is a truly perfect form of a table and pattern mining might be a way to come close to this form.
Asking about the “truth” in patterns, as we do in the title of this article, only makes sense if we assume that patterns are not “real a-priori” knowledge, but theories, that can be tested, supported or falsified. Throughout this paper, we argue that patterns do fall out of the Platonic sky; they do not represent the “idea” of an object or its “true nature”, but offer a theoretical explanation. As such, we use patterns to make sense of the day-to-day practice of design in a pragmatic manner.

Buschmann et al. (2007) state that the process of writing a pattern and the outcome, that is the written document itself, are both part of the pattern: “In most cases moving from one form to another is largely a matter of rewriting and remining the pattern, which is a profoundly creative activity" (p.114). So, describing a pattern is a creative and original act in spite of the "disregard of originality". This is not a contradiction since we are considering two levels here: a pattern is original and creative, but the phenomena we describe with the pattern are not invented. They do "really" exist, whatever this "really" means.

2.2 The structure of patterns and theories

To consider patterns as theoretical constructs is in fact not as original as we have stated before. In his seminal work on patterns, Christopher Alexander repeatedly considers patterns as laws or hypotheses, i.e. as morphological laws:

"Each one of these patterns is a morphological law, which establishes a set of relationships in space. This morphological law can always be expressed in the same general form:

$X \rightarrow r(A, B, \ldots)$, which means: Within a context of type X, the parts A, B, … are related to the relationship r." (Alexander, 1979, p. 90)

More compact: IF: X THEN: Z / PROBLEM Y" (Alexander, 1968)

Because we do not know laws a-priori, we have to formulate hypotheses about the laws. And a network of well corroborated hypotheses or accepted empirical laws is exactly what the kernel of theories is (Bortz & Döring, 2002). Alexander first speaks of patterns as hypotheses that can be tested in "Notes on the synthesis of form" (Alexander, 1964). In "The Timeless Way of Building" (1979) he points out that the distilled invariants of a pattern become empirically vulnerable:

"We can ask ourselves, is it true that this system of forces actually does occur, with the stated context? Is it true that the actual solution, as formulated, really does resolve this field of forces in all cases? Is it true that the precise formulation of the solution is actually necessary: that any entrance which lacks this feature must inevitably have irresolvable conflicts in it, which will communicate themselves to people who pass through it?" (Alexander, 1979, p. 269).

Because each pattern contains a network of hypotheses (the forces that link the context form to the solution form), we think it is adequate to consider a pattern as a theory and not only as an isolated hypothesis. Theories have the function to describe, explain and predict facts and phenomena. They can inspire research in new areas, predict events and provide promising instructions for practical action (Westermann, 2000). These functions of theories are remarkably similar to the general traits of the pattern format:

- a pattern describes the form of **recurrent** solutions and their contexts of applications,
- a pattern **explains the reason** for this fitness in terms of the forces,
- a pattern **predicts** that in another context of a similar kind, the pattern will help to solve the problem,
- a pattern is **instructive** (not prescriptive) for practical design action,
- a pattern is a **three part rule** consisting of IF X (context) THEN Z (solution) BECAUSE of Y (problem as a network of interrelated forces).

2.3 Arguments for considering patterns as theories

In the way Alexander describes patterns they very much look like theories. Why does it still seem odd to consider patterns as theories? The concept of a theory and a pattern are not identical because patterns are specific types of theories.

First of all, patterns are written and documented in a specific way, the pattern format is a text genre different to usual scientific documentation. However, that is only a rhetorical difference.

Second, patterns seek to abstract from concrete design solutions on a medium level of abstraction: “Many in the Hillside Group – and Kent Beck in particular – have established a stigma against ‘going meta.’ We want to grapple with the Stoff of patterns before moving on to generalities and platitudes” (Coplien, 1998). Introducing limits to the extent of generalization or abstraction does not mean that patterns are not theoretical at all. Every abstraction is a loss of information and therefore a loss of “real” substance. We can afford to loose information about surface structure as long as the character of a form category (its essential structure) is preserved. Structure preservation requires that the gestalt (whole form) of patterns must remain perceivable and therefore it limits the level of abstraction.

Third, patterns seem to be very different compared to e.g. the law of conservation of energy. Theories, we think, are usually very general and abstract. For example, the laws of motion apply for all physical objects. A pattern, such as an OBSERVER, only applies in a specific context. However, the OBSERVER pattern should always work if applied in the right context. The context limits the scope of a pattern and therefore reduces the amount of contained information– it does not apply for *ALL situations in the world* but for *ALL situations described in the context*. Such restrictions have impact on the empirical content of a theory.

Fourth, the leading paradigm in the pattern community is that patterns are based on experience. That is, the hypotheses of a pattern are derived from real instances and not “out of the blue”. It is a misconception to think of scientific theories as something invented “out of the blue”. To “invent” a theory means to gather arguments built up from facts, laws and inferences to explain certain phenomena. “Invented” theories use empirical data as well, however, only for testing a theory. Hypotheses are not derived from actual experience but from more general theories.

Scientific explanations differentiate between statistic inductive explanations and causal connection between event A and event B, the latter has to be deduced. Deductive inferences are true due to their structure of reasoning: In a deductive argument, if the premises are true, then the conclusion is true. From the truth of a first sentence (premise) follows necessarily the truth of the second (conclusions). Of course, deduction is fallible as we cannot logically prove all the premises are true.

While Alexander’s work does foresee rational inventions of patterns, the current paradigm in the pattern community is to build a pattern (a theory) upon past experiences, best or good practices. “The empirical nature of patterns suggests that they should be grounded in real examples” (Buschmann et al., 2007). Usually, patterns are inductively inferred from real examples. As we will see, the concept of induction has been critically discussed in the philosophy of science for at least two centuries.

We will approach the nature of discovering theories and patterns in chapter 3. The specific methods of pattern mining will be discussed in chapter 4. Chapter 5 discusses the level of confidence we can put into the methods and mined patterns, suggesting that written patterns often need further corroboration. The final chapter discusses the difference between general (theoretical) designs and specific (implemented) designs and the required skills in creating a design.

3. Induction, Deduction, Abduction

Looking at patterns as specific theories means that we do not have to invent new standards for testing and justifying patterns. Instead, we can rely on tried and tested methods – the patterns of scientific inquiry and empirical methodology. The objective of scientific inquiry is to create knowledge. Knowledge, to-know-that and to-know-how, can be defined as justified beliefs about facts, models and theories about the world (Schnädelbach, 2002). Likewise, a pattern expresses generative rules (laws or regularities) for the design of artefacts. We believe that these rules are tried-and-true and therefore justified.

From an empirical perspective, all knowledge must be based on experience. The phenomena of interest have to be observed and measured precisely and then we can try to postulate theories that summarize and explain the facts. Hypotheses are the result of inductive inference, coming from the specific to the general, from the concrete to the abstract. Inductive inferences are potentially truth extending and are the only strategy to find new insights.
On the opposite site, rationalists claim that there are significant ways in which our concepts and knowledge are gained independently of sense experience. Hypotheses are not the result, but the origin for empirical inquiries which test them.

In a deductive argument, the conclusion cannot extend the information given in the premises. For this reason, inductive inference seems to be more attractive and inductive empiricism is indeed the approach we apply when we mine for patterns. Digging for “nuggets of wisdom” seemingly means to discover what is already out there. However, induction comes with some problems as we will see.

3.1 Induction

In science, enumerative induction is the process of inferring from a number of observed positive cases the properties of new cases. This can be done for a specific case (extrapolation: this design has worked several times, so it will work in this case as well) or for all possible cases (generalisation: this design has worked several times, so it will always work in similar contexts).

Strong Induction (Generalisation)	Weak Induction (Extrapolition)
A_1 is a B_1. A_2 is a B_2. A_n is a B_n. Therefore, all As are Bs.	A_1 is a B_1. A_2 is a B_2. A_n is a B_n. Therefore, the next A will be a B.

Induction is also eliminative (Westermann, 2000): in the process of extrapolation or generalizing it is also assumed that there are no other forces which (generally) influence the observed relations. Hence, we can say that mining design patterns is enumerative and eliminative induction. Although a new specific design task will indeed introduce new aspects to the context, new forces and a modification of the specific configuration, we are assuming that the essentials captured in the pattern do not miss critical forces and do not contain forces that were only incidentally at work in previous design cases.

The benefit of extending the knowledge about specific cases to general cases is at the same time the biggest problem of induction. Because inference by induction extends the information content (e.g. the conclusion contains more than the premise) there is uncertainty about the truth of the conclusion because it is not logically necessary that the cases of the past imply cases for the future, i.e. what has worked in the past does not necessarily work in the future. Induction builds on the assumption that the observed phenomena are uniform and will behave similar in the future. Besides the fact that this uniformity might be of different degrees (e.g. physical objects might behave more uniform than design problems or human behaviour), the assumption that the universe behaves according to the principle of uniformity is a theory that unfortunately builds on induction itself. This problem of lack of rational justification for this principle cannot be resolved (Hume & Buckle, 2007). We can handle this problem pragmatically and accept that we can never be completely sure.

From a constructivist point of view, there is yet another problem with induction. Since meaning is not inherent in the objects and artefacts surrounding us, but rather actively constructed, we can turn any given set of data into a variety of different theories (or patterns). Consider the following example:

E3: “A husband believes that his wife dislikes to be seen with him in public. As ‘proof’ he describes an occasion when they were late for an engagement, and as they were walking briskly from their car, she kept staying behind him. ‘Not matter how much I slowed down,’ he explains, ‘she always stayed several steps behind me.’ ‘That is not true,’ she retorts indignantly. ‘No matter how fast I walked, he always kept several steps ahead of me’” (Watzlawick, pp.62-63)

Whenever we capture regularities in the data we observe, we use personal “punctuations” to determine cause and effect, beginning and ending of a situation. Furthermore, patterns not only build on the observable structure, but on further assumptions about the functional or even causal relations between objects. Functional arguments like “X works because of Z”, “Z is there because it causes X” always contain an explanatory level that is not inherent in the data. Software design patterns are not just the recurrent class diagrams (the data) but also the forces (the reasoning). This indicates that patterns are theory-laden. Just like we cannot be sure who causes the maritial problems in the example given above, we cannot be sure if the pattern we observe is actually truly a whole part of good design – or if there even is a pattern outside our perception.

3.2 Deduction / Falsification

Since we cannot proof whether a theory (or a pattern) will work in the future or in each and every case, verification remains impossible. Theories cannot be empirically verified (as positivists had thought) but only be tested. If they pass the test we can put more confidence into a theory. If theories consequently fail a test, we should reject a theory. This is the core idea of Popper’s critical rationalism (Popper, 1972). To resolve the problem of induction Popper argued that science does not rely on induction, but exclusively on deduction, by making modus tollens the centerpiece of his theory. It has the following argument form:

If P, then Q.
¬Q
Therefore, ¬P.

Science is gradually advanced as tests are made and failures are accounted for. To be tested, a theory (respectively the contained hypotheses) must be falsifiable. That is, it must be possible that if a theory is tested, it can fail.

The same should be true for patterns. We should not assume that patterns are true or that we could provide proofs for patterns. A pattern based on proven design does not imply that the pattern itself is proven. Rather the proven designs provide evidence (not proof) for a pattern. What qualifies a specific design to be proven is another question. However, even the strongest tests for existing designs do not imply that we have generalized the pattern appropriately. And even if so, we cannot be sure that it will work in the future.

Corroboration of theories should not rely on past data because there is the danger of making up a theory that just fits to the previous cases. Patterns that are mined based on “real stuff” must necessarily be formulated in a way that they account for previous cases. But do they hold for future cases? This we do not know. The degree of corroboration is higher if we test a pattern for new cases and we must allow the option that patterns can fail. That makes a pattern empirically vulnerable: if we apply a pattern in the right context but it does not solve the present problems it is indeed falsified. That patterns can fail in principle is a good thing because it is the only way to test them. To which extent a theory can be tested depends on its empirical content (see section 5.3). To which extent a theory is actually tested and evaluated is another question (see section 5.5).

From Popper we have learnt that there is no absolute verified truth in scientific theories. The same applies to patterns. There are no “verified patterns”, only patterns with more or less good evidence.

3.3 Abduction / Retroduction

Abduction is to look for a pattern in a phenomenon and suggest a hypothesis. Unlike deduction and induction, abduction is a type of critical thinking rather than symbolic logic. The objective of abduction is to determine which hypothesis or proposition to test, not which one to adopt or assert. Abductive reasoning usually takes the following structure:

X is observed.
Among hypotheses A, B, and C, A is capable of explaining X.
A is a probable explanation for X.

Simon argues against the “mystical view towards discovery” shared by creative scientists and artist (Simon, 1973). While the falsification method is a strong argument for testing a hypothesis it is not for discovering them. Referring to Hanson’s “logic of retroduction” (Hanson, 1958), he discusses an example of finding patterns in given data. His simple example is the following sequence of letters:

“ABMCDMEFMGHMIJMKLMMNMOPMQRMSTMUVMWXMYZMABMC….

Looking on the sequence, we find that there is a pattern in it:
$n(\alpha)n(\alpha)s(\beta)$; $\alpha = Z$, $\beta = M$

“where ‘$n(\alpha)$’ means replacing a symbol by the symbol next to it on the alphabet, α; ‘$s(\beta)$’ means repeating the same symbol as β; while the expression ‘$\alpha = Z$’ and ‘$\beta = M$’ set the initial values on the alphabets, at Z and M, respectively.”

Simon points out that we can be certain (i.e. verify) that it is a law for the given sequence. However, we cannot be sure whether it is an appropriate law for continuing the sequence (for the uniformity of nature cannot be verified): "…whether the pattern will continue to hold for new data that are observed subsequently will be decided in the course of testing the law."

Finding the law for the sequence, Simon suggests, is released from the problem of justifying induction, because one can consider the process of finding laws without claiming that the discovered law is the unique description, or the most parsimonious possible. *Testing* and *finding* regularities can follow different norms.

Finding laws or law proposals (hypotheses) can be done in an efficient or inefficient way. The inefficient way is what he calls the "British Museum Algorithm" to "honor the monkeys who were alleged to have used it to reproduce the volumes in the British Museum." In other words: this process, the finding of theories out-of-the-blue, is randomly or based on trial-and-error at its best. The algorithm might produce the right law for re-creating the pattern in reasonable time for simple cases. However, if we use a heuristic search algorithm and apply strategies to find the pattern, we might be far better off. The heuristic search algorithm "extracts information from the sequence in order to generate directly an alternative that will work. The difference between the two algorithms is exactly parallel to the difference between solving an algebraic equation by trying possible solutions in some systematic order, and solving it by eliminating constants from the left side, next eliminating variables from the right side, and then dividing through by the coefficient of the variable" (Simon, 1973).

It turns out that this approach, which is also applied in the process of pattern mining, is an appropriate way for finding new hypotheses. To believe that the law is likely to hold true in future cases cannot be said by this method – it offers only a probable hypotheses that must be tested.

4. Methods for pattern mining

The mining of patterns is an attempt to find the regularities and generative rules of design forms:

> "In all these cases, no matter what method is used, the pattern is an attempt to discover some invariant features, which distinguishes good places from bad places with respect to some particular system of forces. [...] It is in the invariant behind the huge variety of forms which solve the problem. There are millions of particular solutions to any given problem; but it may be possible to find some one property which will be common to all these solutions. This is what a pattern tries to do." (Alexander, 1979, p. 260)

In „The Timeless Way of Building", Christopher Alexander names the following ways to find patterns:

- Observation and analysis of good examples
- Analysis of bad examples and inference of good solution
- Inference by pure argument

All three methods can be used to find invariant features that discriminate good from bad designs. They refer to different approaches in finding theories:
Induction: The observation and analysis of existing cases is inductive inference.
Inductive-Deductive: Analysing the commonalities of bad examples is an inductive inference which is followed by deductive inference of a working solution – the lessons learnt.
Deductive: Inference of good solutions by pure argument only based on theoretical assumptions.

While the deductive approach was the rational component of *The Program* in “Notes on the synthesis of form” (Alexander, 1964), Alexander later sees deduction only appropriate for occasional cases. Patterns are usually inferred from experience and not deduced from theories. An example for a theoretical pattern is the first pattern Alexander described in “A Pattern Language” (1977) on WORLD GOVERNMENT. This pattern has never been applied and whether it works or not cannot be tested.

In the pattern community, the inductive approach is the agreed paradigm. Patterns are derived from practical experience and not deduced from theories – or even supported by them. Discovering a pattern is called pattern mining. This metaphor emphasises the analysis of existing design structures and the implicit knowledge of experts. The process of pattern mining reveals “nuggets of wisdoms” from the structure and form of artefacts and the decision making of their creators. To expose the invariant structure and discriminate it from the surface structure (non-essential features) is the main task of pattern mining. It is important to point out, once again, that the generalization from single cases, the reasoning about causalities of working forms and the judgement between relevant and irrelevant features is speculative. The pattern itself is a theory. Or, as Brad Appleton (2000) puts it: „A pattern is where theory and practice meet to reinforce and complement one another, by showing that the structure it describes is useful, useable, and used!“ Typical methods for the inductive inference of patterns can be found in qualitative research and comprise techniques such as observation and analysis, retrospective, expert interviews, focus groups.

Kerth & Cunningham (1997) and DeLano (1998) both name the following methods:

Introspective approach / individual contribution: self observation and analysis of one’s own projects, which processes and designs have been successful or not. Since this approach explains the results rather than the internal feelings and thoughts of a designer, we suggest to call this the **retrospective approach** rather than an introspective approach.

Social approach / secondary contribution: Observation of the environment and the behaviour of its agents, interviews with experts who explain their own experiences or patterns. A methodology to extract patterns from case studies has been developed by Mor & Winters (2008)

Kerth & Cunningham (1997) mention additionally:
Artifactual approach: Observation and analysis of project results. Many of Alexander’s architectural patterns have been induced by this method: by studying existing buildings. Many software design patterns are developed using this approach (Buschmann et al., 2007).

DeLano (1998) mentions additionally:
Pattern Mining-Workshops: Focus groups are used to collect, categorize and summarize the experience of experts. Such discussions often show that there are different views on the same issues which can be harmonized in the course of discussion.

All of the afore mentioned approaches use inductive inference to gain new findings from the field. This is typical for qualitative research (Flick, 1998). The shepherding process (Harrison, 2006) and the writer’s workshops (Gabriel, 2002), too, are qualitative methods that primarily ensure the quality of the pattern description. Both methods help to find errors, gaps, and ambiguousness in the description. Usually the shepherd and workshop participants are also familiar with the subject and can sometimes support the pattern with additional cases and variations of the pattern. Hence, the pattern writing process is also a method of pattern mining because it reveals new facts either by adding additional experiences or asking the authors to express more of their implicit knowledge in the written pattern. Since the patterns are presented to other experts of the field, the workshop is a first test for the patterns because workshop participants can oppose to the content of the pattern description.

5. Confidence and Corroboration

The previous chapter has shown that there are various methods for pattern mining. The pool of qualitative research methods may offer additional ways to mine patterns. Unfortunately, not every pattern paper reports about which methods have been applied and in which environments the cases (the pattern's substance) have been found. However, such information is critical to judge the confidence and scope of a pattern.

Alexander has used asterisks to indicate his level of confidence into his patterns and some pattern authors have adopted this style. But this is a rating done by the authors and likely to be biased. In particular, the scope of a pattern's context may be limited to the experiences and attitudes of the authors. Information about the applied mining methods and the mining field could support less biased confidence. The sections in this chapter will discuss which meta-information can supply less biased indicators for confidence and corroboration.

5.1 Levels of objectivity

Both the mining ground and the chosen method have implication for the objectivity of a pattern. There is a difference between a single author reporting her/his patterns or the outcomes of a group discussion. Likewise the range of domains and contexts in which a pattern has been observed is critical to its general applicability. For example, if a pattern has been observed multiple times in Java programs, does this necessary imply that it will work for C++ code as well? Without having observed or tested it, one cannot really (empirically) tell.

Since the creation of categories depends on the objects known and the properties considered, we can hardly speak of objective categories. Note that objective does *not* mean closer to the truth but that judgments (deciding whether one configuration is of a specific category or not) are inter-subjective. There may be objective laws how people perceive whole forms and construct their mental categories and patterns, e.g. the gestalt laws (Wertheimer, 1938; Goldstein, 2009), Alexander's 15 fundamental properties (Alexander, 2000a, 2000b) or schema theory (Kohls & Scheiter, 2008)[1]. While in these cases the process may be universal, the results are not. The traces of perception, experience and manipulation of mental structures are unique history for each individual and therefore result in personal views of the world.

If some objects do not vary very much, e.g. a soda bottle, it is easier to denote a category people agree on. If there are only some commonalities it gets harder, e.g. a lecture can have many forms. People not only have different ideas about how an ideal lecture looks like, they might even debate if an extraordinary lecture is a lecture at all. The functional forces documented in a pattern are even less objective because different individuals may care for different functions and values. Which functions are critical and relevant depends on the

---

[1] Note that Alexander's concept of wholeness is rooted in Gestalt theory (whole = gestalt) and his 15 fundamental properties are very similar to some gestalt laws. Gestalt laws try to explain how we perceive whole forms. Schema theory is connected to Gestalt theory as well. It focuses on how forms and patterns are stored as mental structures and influence how we respond to a new stimulus. One critique of Alexander's 15 properties should be that it omits the gestalt law of familiarity – the echoes and symmetries of the past.

interests of the individual. The least objective properties are those of beauty. Of course, there are objects most people consider as beautiful, e.g. rainbows or water lilies – but just as well, they might consider a picture of these objects as mere kitsch. In the aesthetics of Kant, Hegel, Wittgenstein and others (Majetschak, 2007), beauty and wholeness depend on our familiarity to a category. Somebody who is competent in the domain is capable of giving reasons for her/his aesthetical judgement. For example, we cannot only say that an OBSERVER is a beautiful design (in specific contexts) but also give reasons why it is the right thing. It is not just a solution but a good solution. A beautiful solution is one that does the right thing and fits well within our own aesthetic categories. In the case of the OBSERVER, being a programmer is necessary to see the inherent beauty of this pattern.

Individual categories and normative categories (e.g. ethical values) are counter arguments for objective and true categories, because such categories can change and have changed over time. Many norms in software design appear to be reasonable conventions, e.g. maintainability, robustness, performance, memory optimizing, etc. These conventions are implicitly agreed upon in the community of practice. Teaching practice, apprenticeship and also software design patterns make these categories explicit, and thereby inter-subjective agreement becomes more likely.

5.2 Justification for induction

Individual skills, attitudes, beliefs, volition and prior knowledge are as much part of the context of a pattern as any other environmental aspect. People may prefer different styles of architecture, painting, or coding. What people prefer and value has changed over time and often depends not only on regional cultures but on peer-groups and communities. Hence, the cultural setting and background is critical for the contextual validity of a pattern.

This calls for more transparency in pattern mining. To evaluate a pattern, it is crucial to learn in which environment a pattern was mined, which attitudes and beliefs the author held, and how many different views and artefacts have been examined to come up with the pattern. To assume that one designed form will work in other cases and for other people very much depends on how stable the past cases have been and how similar the new cases and cultural settings are.

Again, we can learn from the theory of science, since here we find established factors which are critical for the justification of induction. Westermann & Gerjets (1994) name four factors that influence the justification for inductive inference to new cases: sample size, validity and variation of previous positive cases and the similarity to a new case.

Transferring these justifications to the realm of patterns we can say that a pattern is better justified if there are **more positive examples** it builds on. Also, the number of negative examples (cases where the claims of the pattern are false) should be low compared to the positive examples. The **validity of the cases** depends on the objective perception of whole forms (which is critical as the last chapter has shown). Another justification comes from the **variation of cases**: if there are a lot of different cases in which the pattern succeeded, it is more likely to be a justified plan for new designs. For instance, a pattern that claims to have a wide scope of application is better justified if it has actually worked in different settings and for different domains. The pattern is also strengthened if the negative cases are similar (e.g. the pattern does not fail generally but under certain conditions – such conditions could change the described context in the evolution of the pattern). The justification for a designer to choose a particular pattern for her/his design is better if the design is **similar to the positive**

cases, and it is lower if it is similar to negative cases. This implies that the designer shares the same intents, values and attitudes the pattern is based on.

5.3 Empirical vs. logical content

The extent to which we can test a pattern depends on its empirical content. The confidence we can put into a pattern does not only rely on the number of cases it is based on but also on how much a pattern claims. If a pattern claims to work in many different cases (unspecific context) it must be testable in many different cases. If a pattern gets very detailed and specific in its solution it claims more than a pattern that lets unanswered a lot of question. The level of specification of a pattern measures its content, e.g. what is contained in a pattern. We can distinguish between the logical and empirical content of theories or patterns respectively.

The two compounds are opponents: A theory with a low amount of logical content has a high amount of empirical content. A form (situation or phenomena) that is fully specified logically contains exactly one case. On the other hand, forms that are not specified at all contain any kind of configuration - the logical content is maximized because logically all cases are contained. Let us consider the logical and empirical contents for patterns.

The more constrained a context is, the lower the empirical content is.
If there are a lot of conditions conjunct in the context, the logical content that constraints the applicability is high. Hence, the empirical content is low for there are fewer cases in which the pattern can be used. The context is very specific.

To see how the constraints logically contained in the context reduce the number of empirical applications, consider a pattern that applies only for programming language X in the application domain of Y. It has a lower empirical content than a pattern that applies to all languages and to all domains.

The more general a context is, the higher the empirical content is.
If there are a lot of alternatives in the context (e.g. a pattern works under these conditions OR that conditions), the logical content that constraints the applicability is low (the context is less specific), whereas the empirical content is higher. There are more cases in which the pattern can be used and empirically tested.

The more specific a solution is, the higher the empirical content is.
Likewise, the logical and empirical content of the solution depend on each other. If you precisely describe what design options are allowed, should be avoided, or have to be done, then you are explicitly limiting the number of possible solutions. The more constraint a design space is, the fewer designs are contained in it – therefore its logical content is low. Because of the precise and detailed description, the empirical content is higher. There are more things expressed about the design and more things can be tested (and falsified) accordingly.

Consider this example: If you have a pattern for a method in online education that tells you one possible time span of running it, the empirical content is low. For example if you say that an ONLINE TRAINING can adequately last 30 minutes then you have exactly one time value to test. Its empirical content is low because it offers only one test case. Its logical content is high because it does not claim anything about trainings that last 29 or 31 minutes. Hence, for a training that lasts 31 minutes it logically includes both case, the one in which the time span works and the one in which it does not. If you extend the claim in the solution statement and say that a time span between 30-45 minutes is adequate, you increase the empirical content.

You make the solution more specific by saying that 30 AND 31 AND 32 … AND 45 minutes will work. Hence, you logically exclude the case that 31 minutes are *inadequate* for a training, claiming that it is always *adequate*. By giving a range of possible values we extend the assumption. If it is true that all the options are equally valid solutions then this is a desirable extension[2].

The less specific a solution is, the lower the empirical content is.

If the solution is less specific and suggests different paths without saying which one will actually work, its logical content is higher because it contains alternative forms. But it does not tell you which of the alternatives will actually work. To say that A *or* B will work logically means that it is left open which one actually does (to say that both forms work would logically be expressed as A *and* B work, thus, the statement would be more specific not less).

To understand the implication, imagine that you have a specific design problem and somebody tells you that **one** of the Gang of Four or POSA design patterns will help. The logical content of this solution proposition is high since all the patterns are included. The empirical content is low because it does not tell you which one will work.

A more extreme example is Joseph Bergin's pattern "Do the right thing" which makes fun of the idea of having a universal pattern. Its solution is very general and unspecific: "Do the right thing. Make the bad thing better." Its logical content is maximized. Doing the right thing could be anything. In fact, if you consider *anything*, then *something* will work. Therefore its empirical content is zero. We know in advance that because of the logical structure the statement is always True. It is a tautology! There simply is no way to falsify this pattern because it includes all possible things to do that *might* make things better. The point is, it does not tell us **which** of the many things to do.

Testing the empirical hypotheses

So far we have only talked about context and solution. Since we have seen that a pattern is not just a simple hypothesis (*if* context *then* solution) but a network of hypotheses that explain the forces that cause the fitness between context and solution, these hypotheses count for the content as well. Each force tells something about the context and problems, and each force can be falsified empirically. That is, we can test whether all the forces actually exist in a given context and whether all the force are actually resolved by a given solution.

It is important to notice that we can test all the claims and forces empirically, but we cannot do this in isolation. We cannot test a single force or a single design variable ceteris paribus. The reason is that there are interdependencies between the form variables. If one puts a different weight on a single force (e.g. changing a force ceteris paribus), the complete design may have to change. We can test a pattern only as a whole. As a consequence each test must include all the statements that are contained in a pattern. Usually this includes a lot of implicit tests and the amount of empirical content is indeed critical to justify the "Rule of Three" as statistically significant evidence for prior positive cases.

---

[2] Of course, the exact bounds for adequate time spans of online trainings are fuzzy and not precise. Bean counting the minutes was only to demonstrate how the logical and empirical content change if the range of specified values is modified.

5.4 The Rule of Three

The “Rule of Three” informally suggests that there should be at least three known uses. A singular solution is just a design, two occurrences might be random, whereas the third occurrence makes a solution a “pattern”. Of course, the recurrence of a design configuration could still be random. Then, why do we accept the “Rule of Three” heuristically as sufficient evidence to talk of a pattern? The question that we have to ask is: how likely is the invariance of successful design configurations just random outcome? The answer is that the statistical significance depends on the logical and empirical content of a pattern. Let us assume that we have a number of different design objects codified in binary form by representing various design variables and relations as binary strings. An example is shown below:

0011010100010001000001001   0010101010111111101011101   1101000010101000111111001
1111000111110111110001001   1101000010101100111010111   1000010110101110101000001
1101000010101111100011100   1111011000011111000100011   1111110111011011100011111

When we are looking for patterns in the design of objects, we usually look for recurrences[3]. The pattern “1001” appears three times[4] at the end of a string, the pattern “1101000010101” appears 3 times at beginning. How likely is it that these recurrences are just random? For the “1001” it is quite likely because if you take only four binary variables into account, roughly every eighth (2^4 =8) object could have this configuration by chance. Its logical content is huge because there are many objects in the world having randomly this configuration. Its empirical content is low, because there is not much we can test and learn from. The other pattern, however, is a string of 14 digits. Thus, it should recur by chance only one out of 2^{14}=5096 times[5]. In our example it appears three out of nine times. This could still be a recurrence by chance, but it is less likely. The “Rule of Three” is significant in most cases because design patterns usually have a high amount of empirical content in their solution parts. Inseparable design variables suggest that a single test consists of multiple fallible statements. It is just not very likely that in design objects tackling the same problem similar configurations occur again and again by chance.

This is not an advocacy for finding absolute truth through inductive reasoning. We are in line with the critical view Hume has put forward in the middle of the 18th century in his work “An enquiry concerning human understanding”[6]. If we take any regularity as a pattern, claiming a causal connection, we may end up like Aristotle. He noticed that mice were commonly found in barns where grain was stored. He thought that the mice grew from the grain and hay, and he coined the term “spontaneous generation”, the hypothesis that living organisms arise from nonliving matter. As a matter of fact, he published a recipe that anyone could use to grow their own mice: darkness + hay + grain = mice.

The “Rule of Three” can make a pattern significant but not necessarily plausible or true. But this is a problem shared by any statistical method. As such the “Rule of Three” is no better or worse. The difference is that due to the complexity of statements tested at once, fewer cases are needed to make the invariance significant.

---

[3] More precisely we are looking for whole forms or perceivable coherent gestalts that recur.

[4] To simplify, we do not distinguish between context and solution. Rather, we only consider the solution forms and assume that all objects are satisfying solutions to the same design problem.

[5] Note that the chance to find *any* pattern in a huge number of objects is much higher. This is similar to the birthday game in which you need only roughly 20 people in order to find two persons with the same date of birth. Still, the reoccurrence of *complex* patterns is much less likely than of *simple* patterns.

[6] Online edition: http://18th.eserver.org/hume-enquiry.html

5.5 Testing a hypothetical pattern

As a matter of fact, if we consider a limited set of objects (as in the previous example) it is not even likely that a specific pattern (logically only a few configurations are allowed) re-occurs two times by chance. So, actually, three recurrences is the minimum number of cases required to test an induced pattern (from two occurrences) at least once. Hence, the “Rule of Three” sometimes implicitly includes a test of a pattern. For example, some of the patterns for interactive information graphics (Kohls & Windbrake 2006, 2007, 2008a, 2008b) were mined using the artifactual method by investigating various multimedia applications (Kohls & Uttecht, 2009). A first recurrence of interaction forms qualified the form as a pattern candidate. Another occurrence qualified it as a pattern. In this case, the pattern candidate was the hypotheses (an assumed invariant form) and another occurrence was a first corroboration. This works also with other mining methods such as individual contributions: Very often, we draw from our experiences and think “this could be a pattern” (we have a hypothesis). Then we see the assumed pattern applied in another design and think that it is indeed a stable pattern. We have our first qualitative corroboration.

Of course, it is desirable to have patterns and the respective pattern descriptions tested more systematically. Pattern descriptions usually include a section entitled “known uses”. However, we never learn whether the pattern was induced from these cases or whether (some) of the cases have been used to test or support the pattern.

Furthermore, it is of interest to evaluate the quality of a pattern, whether there are successful uses of the pattern description, i.e. its actual application to new cases by third parties. It would be beneficial to learn about failures as well. Unfortunately, authors usually do not learn about the application of their patterns. As a first step, pattern authors should encourage readers to provide feedback about the application of their patterns. Summaries of the feedback – positive and negative – should be part of the pattern description, provide evidence or counter-evidence for the quality of a pattern.

6. Conclusion: Is it a science or an art?

If science is about the nature of things, then patterns certainly belong to science. In the case of patterns, the things or objects of consideration are artefacts and practices of creating the artefacts. Hence, patterns are a way to investigate the "science of the artificial" (a term coined by Simon, 1969), or the nature of artificial objects. However, patterns are not about science only. The scientific component of the pattern approach is the mining of invariants in both designs and the design processes, and to investigate the reasons and causes for specific forms (what forces a form to take its shape to serve in a specific context). But to actually run the process and to create the artefact is a matter of skill and craftsmanship, and this is where the pattern approach has its artistic component. That there is a difference between the general rule and the specific rule is pointed out by Aristotle already (Schnädelbach, 2003). Knowing the rules of painting does not imply that you can actually paint. On the other hand, knowing a finite set of rules to create paintings (or any artefact) does not mean that you cannot originally create new forms by applying the rules, nor does it exclude that one can find new rules that work.

Whether patterns are indeed useful and their descriptions are usable is independent from the corroboration and inter-subjective reality of a pattern. A pattern can be obvious, but so simple that it is not useful. It also can be too complex, only useable in very specific situations, or hard to understand. Such factors belong to the pattern descriptions and not to the design categories they capture. They can be studied and tested independently of the claims contained in the pattern. The “validity” of a pattern is a necessary but not a sufficient condition to make a useful and relevant documentation. The quality of a pattern highly depends on its usefulness. It is useful when it is corroborated and helps a designer solving important problems in recurrent situations. Still such a pattern could be pure fiction. We have several times referred to Newton’s laws. While these “laws” are useful approximations, according to the theory or relativity the laws of physics are different all together. However, you can observe the differences only in extreme situations – Newton’s laws work in most but not in all contexts. Therefore, they are very useful without being true. Likewise a pattern can be very useful without being true. Maybe the notion that every pattern tells a story should be taken more literally. But from a pragmatic point of view what matters is not the truth but the usefulness of the patterns in solving important problems.

We hope to have shown that patterns are theories and not “real stuff”. Rather, they are inductively inferred from “real stuff”. While we have argued that induction and abduction are appropriate methods to derive hypotheses, we must stress that this does only explain the designs of the past and that alternative explanations can always be given. Therefore, the hypothetical explanations and design predictions have to be tested in order to strengthen a pattern. In fact, a good designer tests a pattern before s/he uses it, e.g. decide whether a pattern actually fits the situation at hand. Patterns do not release the designer from the responsibility to think about the design. For the pattern community, it is important to realize that each application example only provides evidence and not proof that a pattern actually works. Proofs can only be provided for pure mathematical inference and logical deduction.

As an outlook, we suggest that pattern papers should not only include the pattern descriptions but also give more space to document:

- The mining ground (variation of cases)
- The mining methods (validity of cases, confidence, objectivity)
- Which known uses induced a pattern? (sample size and variation of cases)
- Degree of corroboration: Which known uses were deduced from the pattern and succeeded? (successful application to similar cases as evidence)

Acknowledgement

The authors are very grateful to the help and support of our shepherd Linda Rising. Linda has done what a good shepherd does: she has stopped us when we were running into wrong directions and – more frequently – were running too far off topic. She has taken care of us and fed us with valuable input and tips.

References

Alexander, C. (1964). Notes on the synthesis of form. Cambridge: Harvard University Press.
Alexander, C., Ishikawa, S., and Silverstein, M. (1968). A pattern language which generates multi-service centers. University of California, Berkely: Center for environmental structure.
Alexander, C. (1979). The Timeless Way of Building. New York: Oxford University Press.
Alexander, C., Ishikawa, S., and Silverstein, M. (1977). A pattern language: towns, buildings, construction. NewYork: Oxford University Press.
Alexander, C. (2002a). The nature of order, Book 1. The phenomenon of life. Berkeley, Calif: Center for Environmental Structure.

Alexander, C. (2002b). The nature of order, Book 2. The phenomenon of life. Berkeley, Calif: Center for Environmental Structure.
Appelton, B. (2000) Patterns and Software: Essential Concepts and Terminology. http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html. (accessed July, 2009)
Berkun., S. (2007). *The myths of innovation*. Sebastopol, CA: O'Reilly.
Booch, G. (1998). Patterns. In Rising, L. The Pattern Handbook. Cambridge: Cambridge University Press.
Bortz, J., & Döring, N. (2002). *Forschungsmethoden und Evaluation: Für Human- und Sozialwissenschaftler*. Berlin [u.a.]: Springer.
Buschmann, F., Henney, K., and Schmidt, D.C. (2007). Pattern-oriented software architecture. Volume 5: On patterns and Pattern Languages. West Sussex: John Wiley & Sons.
Coplien, J. O. (1996). Software Patterns. New York: SIGS Books.
Coplien, J.O. (1998). Setting the Stage. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.
Coplien, J.O. (1998b). Space: The Final Frontier. C++ Report 10(3), pp. 11-17. March 1998
Corfman, R. (1998). An Overview of Patterns. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.
DeLano, D. E. (1998). Patterns Mining. In Rising, L. (1998). The Pattern Handbook (pp.87-96). Cambridge: Cambridge University Press.
Flick, U. (1998). *An introduction to qualitative research*. London: Sage.
Foote, B. (1997) In Martin, R.C., Riehle, D. & Buschmann (eds.). Pattern Languages of Program Design 3. Addisson Wesley.
Gabriel, R. P. (2002). *Writers workshops and the work of making things: Patterns, poetry*. Boston [Mass.]: Addison-Wesley.
Gabriel, R. P. (2008). Writers' Workshops As Scientific Methodology. http://dreamsongs.com/Essays.html
(accessed August, 2008)
Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley.
Goldstein, E. Bruce. (2009). *Sensation and Perception*. Wadsworth Pub.
Hanson, N.R. (1958). Patterns of Discovery. Cambridge.
Harrison, N.B. (1998). Potential Pattern Pitfalls, or How to Jump on the Patterns Bandwagon Without the Wheels Coming Off. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.
Harrison, N.B. (2006). The Language of Shepherding. In Manolescu, D., Völter, M., and Noble, J. (Eds.). Pattern Languages of Program Design 5. Boston: Addison-Wesley.
Hume, D., & Buckle, S. (2007). *An enquiry concerning human understanding and other writings*. Cambridge texts in the history of philosophy. Cambridge: Cambridge University Press.
Kerth, N.L., Cunningham, W. (1997): Using Patterns to Improve Our Architectural Vision. IEEE Software 14(1), Seite 53-59.
Kohls, C., & Scheiter, K. (2008). Mental acquisition of design patterns.. Proceedings of the 2008 conference on Pattern Languages of Programs (PLoP). Nashville, Tennessee: ACM.
Kohls, C. & Uttecht, J. G. (2009). Lessons learnt in mining and writing design patterns for educational interactive graphics. Computers in Human Behavior.
Kohls, C., & Windbrake, T. (2007). Moving objects. http://hillside.net/europlop/europlop2007/workshops/F4.pdf.
Kohls, C., & Windbrake, T. (2006). Towards a Pattern Language for Interactive Information Graphics. http://hillside.net/plop/2006/accepted_papers.htm.

Kohls, C., & Windbrake, T. (2008a). Turning me on, turning me off (Writer’s Workshop version). 13th European Conference on Pattern Languages of Programs, 2008. http://hillside.net/europlop/europlop2008/submission/schedule.cgi.
Kohls, C., & Windbrake, T. (2008b). Where to go and what to show: more patterns for a pattern language of interactive information graphics. Proceedings of the 2006 conference on Pattern languages of programs (pp. 1-11). Portland, Oregon: ACM.
Majetschak, S. (2007). *Ästhetik zur Einführung*. Hamburg: Junius.
Mor, Y. & Winters, N. (2007): Design approaches in technology enhanced learning. Interactive Learning Environments, 15(1): 61-75.
Noble, J. (1998). Classifying relationships between object-oriented design patters. Australian Software
Popper, K. R. (1972). *The logic of scientific discovery*. London: Hutchinson.
Rising, L. (1998). Design Patterns: Elements of Reusable Architectures. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.
Rising, L. (2007). Understanding the Power of Abstraction in Patterns. IEEE Software. July/August 2007.
Schnädelbach, H. (2002). *Erkenntnistheorie zur Einführung*. Zur Einführung, 268. Hamburg: Junius.
Schümmer, T., & Lukosch, S. (2007). *Patterns for computer-mediated interaction*. Wiley series in software design patterns. Chichester, England: John Wiley & Sons.
Simon, H. A. (1969). *The science of the artificial*. Karl Taylor Compton lectures, 1968. Cambridge: M.I.T. Press.
Simon, H.A. (1973). Does Scientific Discovery Have A Logic? Philosophy of Science. 40. p. 471- 480
Vlissides, J. (1997). Patterns: The Top Ten Misconceptions. Object Magazine. March Issue.
Wertheimer, M. (1938). Laws of organization in perceptual forms. London: Harcourt, Brace, and Jovanovitch.
Watzlawick, P. (1976). *How real is real?: Confusion, disinformation, communication*. New York: Random House.
Wertheimer, M. (1938). Laws of organization in perceptual forms. London: Harcourt, Brace, and Jovanovitch.
Westermann, R, & Gerjets, P. (1994). Induktion. In T. Herrmann & W. Tack (Hrsg.) Enzyklopädie der Psychologie: Themenbereich B, Seire I, Bd. 1, Methodologische Grundlagen der Psychologie. (S. 428-472). Göttingen: Hogrefe.
Westermann, R. (2000). *Wissenschaftstheorie und Experimentalmethodik: Ein Lehrbuch zur psychologischen Methodenlehre*. Göttingen [u.a.]: Hogrefe, Verl. für Psychologie.
Wittgenstein, L (1922). Tractacus Logico-Philosophico. London: Kegan Paul, Trench, Trubner & Co.
Wittgenstein, L. (1953) Philosophische Untersuchungen, Suhrkamp (2008).
Whitehead, A. N., & Russell, B. (1925). *Principia mathematica*. Cambridge [Eng.]: The University Press