Is that true…? - Thoughts on the epistemology of patterns

Christian Kohls
Knowledge Media Research Center
Konrad-Adenauer-Str. 40
72072 Tuebingen
Germany
c.kohls@iwm-kmrc.de

Stefanie Panke
University of Bielefeld
Universitätsstraße 25
33615 Bielefeld
Germany
s.panke@iwm-kmrc.de

ABSTRACT

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 of 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 scientific procedures. This justification is needed to establish the pattern approach as a scientific methodology beyond the scope of the pattern community. For the pragmatic pattern practitioner, e.g. users and authors of patterns, this paper encourages the critical reflection on the pattern concept. Patterns are not tried-and-true per se, just like theories they have to be subjected to empirical tests. 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.

Categories and Subject Descriptors

D.2.10 [Software Engineering]: Design – Methodologies

General Terms

Design, Human Factors, Documentation, Theory

Keywords

Design Patterns, Theory of Science, Epistemology, Pattern Mining, Evidence, Methodology

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 (i.e. the discovery or explanation of a pattern) 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? In fact, a pattern should not report on surface properties but rather "capture hidden structure" at a "suitably general level" [12]. To reveal such previously unreported structure certainly is a creative act. 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. In the same way, "the most important patterns capture important structures, practices, and techniques that are key competencies in a given field, but which are not yet widely known" [12].

However, not all pattern descriptions contain new findings. If somebody describes the OBSERVER pattern once again, this can hardly be called scientific progress – unless it contains new properties of the OBSERVER pattern which have not been covered before. Research in other fields can also contribute to the content of patterns. As Buschmann, Henney and Schmidt [11] 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 to inspiration by scientific progress, for instance the discovery of new materials, new procedures or new findings in human-computer-interaction.

Also, each new implementation of a pattern in an appropriate context is a test whether the forces are actually resolved by the pattern or not. Such a test can succeed and provide evidence for the "truth" of the pattern. A test can also fail and potentially falsify the claims of the pattern. If a pattern constantly fails, it must be rejected. If a pattern continues to help those designers who apply it in the right situations and in the right manner, the pattern gets more reliable.

However, such corroboration must not be confused with ultimate truth. It is just a level of confidence we can put in those patterns that are helpful to the practitioner. And while patterns are all about practice we should not forget that they are of a theoretical nature nevertheless. In as much as physical theories are about the general physical laws of the universe, theories of practice are about the general rules of creating specific forms. 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. We might have seen "a force at work" but we have never seen a force directly. Hence, patterns are purely rational assumption, and they need, as any theory does, empirical evidence.

In section 2 we will provide arguments why patterns should be treated as theories about "real" things rather than seeing a pattern itself as "real and true". We will approach the nature of discovering theories and patterns in section 3. The specific methods of pattern mining will be discussed in section 4. Section 5 discusses the level of confidence we can put into the methods and their results, the mined patterns, suggesting that written patterns often need further corroboration. The final section discusses the difference between general (theoretical) designs and specific (implemented) designs and the required skills in creating a design.

2. PATTERNS ARE THEORIES

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 [54], 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 example taken from Watzlawick [52].

E1: 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 randomly; 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.

Likewise Feynman [17] warns us to not only consider positive examples of a hypothetical assumption because this could lead to what he calls "cargo science" (an allusion to the anthropological concept of "cargo cult").

E2: "In the South Seas there is a cargo cult of people. During the war they saw airplanes land with lots of good materials, and they want the same thing to happen now. So they've arranged to imitate things like runways, to put fires along the sides of the runways, to make a wooden hut for a man to sit in, with two wooden pieces on his head like headphones and bars of bamboo sticking out like antennas--he's the controller--and they wait for the airplanes to land. They're doing everything right. The form is perfect. It looks exactly the way it looked before. But it doesn't work. No airplanes land. So I call these things cargo cult science, because they follow all the apparent precepts and forms of scientific investigation, but they're missing something essential, because the planes don't land." [17]

As these examples show, our perception of patterns is open to delusion and what we find depends on 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 have to make sure that the structure of a pattern is logically sound.

Design patterns are prone to the same misinterpretations of reality as the next two examples show.

E3: In our collection of patterns for interactive graphics we mined several button forms to dynamically show and hide graphical elements [34]. Technically there is no big difference between showing/changing a graphic on mouse click or mouse over events. We initially mined one pattern out of it. However, in an experiment we asked teacher students which interactive graphics they classified as similar [31]. Not one of them considered the two trigger events as similar in spite of the same reactions and graphics. Even more alarming, when we wrote the two distinct patterns it turned out that they were appropriate to different contexts and resolved different sets of forces.

E4: Schmolitzky & Schümmer [44] have written an excellent language for supervising thesis projects. However, some of the patterns' instant solutions show a preference of the authors to use wikis whenever possible. For example, their DIARY pattern suggests "The easiest way to implement a diary is to write it as a shared wiki page." We have discussed this over with several educators and most said that an e-portfolio or blog system would

be the most appropriate thing for this particular task. Hence, it is more of a personal view than a definite truth which solution works best.

2.1 A pattern is not the same as its manifesting artifacts

Discussing patterns as theories is not common in the community of practice that mines and documents patterns. On the contrary, there is a widespread argument on being suspicious of theories and traditional science. Coplien [13] stressed that patterns are made of "Stoff – real stuff, not platitudes and theories" and Rising [43] 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 view in the community is the notion of patterns as not being artificial, having emerged over time. Patterns are not "created artificially just to be patterns" [11]. Similarly, DeLano [16] observed: "Patterns are not grown or created. They are present in the artifacts 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" [11]. Patterns capture "well proven design experience" [25], and they are "tried-and-true" [16]. Authors like Schümmer and Lukosch [46] 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" [19]. Gabriel [20] stated "software patterns are about describing what works and has worked well rather than finding new ideas".

Yet, we claim 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. We are not opposing 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 a pattern and the objects that manifest the pattern. In the same way there is a difference between a theory and the objective facts that 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 [8]. 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 [3, p. 261] 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". 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 drawing 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 same 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 truths exists 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 "platonic 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. [11] 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, writing 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 -> r (A, B, …), which means: Within a context of type X, the parts A, B, … are related by the relationship r." [3, p. 90]

More compact: IF: X THEN: Z / PROBLEM Y" [2]

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 [10]. Alexander first speaks of patterns as hypotheses that can be tested in "Notes on the synthesis of form" [1]. In "The Timeless Way of Building" [3] 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?" [3, 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 describe, explain and predict facts and phenomena. They can inspire research in new areas, predict events and provide promising instructions for practical action [55]. 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 necessarily 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.

1. Patterns are written and documented in a specific way. The pattern format is a text genre different from usual scientific documentation. Pattern documentations are written appropriately for a particular target audience. However, that is only a rhetorical difference.

2. As theories about practice, patterns have a very powerful implicit test mechanism – a pattern needs to be useful to the practitioner. They are not about all kinds of practices and social behaviours. Only those phenomena that are judged to be relevant and have a direct practical use are picked out. This selection, however, does not make them any less theoretical. It is just a normative value which patterns are worth to be captured.

3. Patterns seek to abstract from concrete design solutions on a medium level of abstraction: "A pattern is abstract because it approaches the problem at a suitably general level, although the solution may entail details. A good solution has enough detail that the designer knows what to do, but it is general enough to address a broad context." [12]. Every abstraction is a loss of information and therefore makes a thing theoretical. 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. Wholeness sets the limit for abstraction. Once we reach that limit (without going further) we have a perfect theory. Many sciences suffer from going beyond that limit and it is therefore that they seem to be too theoretical. However, if a theory looses its connection to empirical data it is actually not too theoretical but simply wrong or meaningless.

4. 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 stated context. The context limits the scope of a pattern and therefore reduces the amount of contained cases – it does not apply for *ALL situations in the world* but for *ALL situations covered by the context*. Such restrictions have impact on the empirical content of a theory.

5. 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". A theory should always explain and predict a phenomenon of reality. There are indeed different ways how theories are developed. While Alexander's work does foresee rational inventions of patterns, the current paradigm in the pattern community is to build a pattern upon past experiences, best or good practices. "The empirical nature of patterns suggests that they should be grounded in real examples." [11] 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.

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 [45]. 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 side, 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.

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).

Table 1. Strong and Weak Induction

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 [55]: 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 [28]. 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:

E5: "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'" [52, 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. 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. Let us illustrate this for software patterns.

E6: Buschmann, Henney and Schmidt [11, p. 149] use a different punctuation for the ADAPTER pattern than the Gang-of-Four does: "Another example of two patterns hiding within a single pattern description is the case of OBJECT ADAPTER and CLASS ADAPTER, which are both contained within the description of ADAPTER [GoF95] […] Although they share a common intent, their complementary nature is reinforced by a dual narrative that runs through the pattern description". If only one of the two views is correct we have to either reject ADAPTER or reject the complementary pair OBJECT ADAPTER and CLASS ADAPTER. However, if we understand that the punctuations are not real but rather made up by the human mind to find meaning, then we can indeed accept both views.

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 because forces are not directly observable. A recent popular science publication exploring the problem of induction and its consequences can be found in the book "The Black Swan" [49].

3.2 Deduction / Falsification

Since we cannot prove 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 [41]. 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 could fail.

The same should be true for patterns. We should not assume that patterns are true or that we could provide ultimate proofs in a mathematical sense 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 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 artists [48]. 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" [24], he discusses an example of finding patterns in given data. His simple example is the following sequence of letters:

"ABMCDMEFMGHMIJMKLMMNMOPMQRMSTMUVM WXMYZMABMC….

Looking on the sequence, we find that there is a pattern in it:

n(a)n(a)s(ß) ; a =Z, ß =M

"where 'n(a)' means replacing a symbol by the symbol next to it on the alphabet, a; 's(ß)' means repeating the same symbol as ß; while the expression 'a =Z' and 'ß =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" [48].

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." [3, 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 learned.

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" [1], 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" [4] on WORLD GOVERNMENT. This pattern has never been applied and whether it works or not cannot be tested.

In the software pattern community, the inductive approach is the agreed paradigm. Patterns are derived from practical experience and not deduced from theories. 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 [7] 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". As such, patterns are theories of good practices. Typical methods for the inductive inference of patterns can be found in qualitative research and comprise techniques such as observation and analysis, retrospectives, expert interviews, focus groups.

Kerth & Cunningham [29] and DeLano [16] 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 calling 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 [37].

Kerth & Cunningham [29] 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 [11].

DeLano [16] 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 mentioned approaches use inductive inference to gain new findings from the field. This is typical for qualitative research [18]. The shepherding process [26] and the Writer's Workshops [20], 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 in 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 Level 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 and 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 necessarily 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 [53, 23], Alexander's 15 fundamental properties [5,6] or schema theory [30][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 a unique history for each individual and therefore result in personal views of the world.

[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.

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 what 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 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 [36], beauty and wholeness depend on our familiarity with 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 [54] 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 has its own problems 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 current design problem 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 and logical content

The extent to which we can test a pattern depends on its empirical content. The confidence we can put in a pattern does not only depend 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 (broad 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 questions. The level of preciseness 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 small 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.

5.3.1 The narrower a context is, the lower the empirical content is

If there are a lot of conditions conjunct in the context, that means that it narrows the number of cases for which it makes definite statements. For all cases not included in the context we do not know whether or not the pattern works, hence for such cases we have logically two different outcomes (applicable or not applicable). If there are only a few cases for which the pattern claims to be applicable (This pattern can be used if a AND b AND c…) this leaves only a few cases for which we can test the pattern. The context is very narrow.

As an example consider a pattern which states in its context that it can be used for the programming language Java and the domain of banking. That it can be used for banking applications does not say that it cannot be used for other domains as well. It just does not make any claims about it. Let us consider what is logically and empirically contained in that pattern:

a) The pattern can be used for banking domains.

b-1) The pattern can be used for game programming OR

b-2) the pattern cannot be used for game programming.

Only a) can be tested empirically because b) is a tautology: it either works or not in the context of game programming. Logically we have three cases because neither b-1) nor b-2) is excluded.

If we extend the context and claim that the pattern works for both banking domains and game programming it contains:

a) The pattern can be used for banking domains.

b-1) The pattern can be used for game programming domains.

We can now test the pattern in two domains, banking and games. Hence, the empirical content has grown. It claims more and there is more that can be tested. The logical content has shrunken because we have explicitly excluded b-2).

5.3.2 The broader 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 those conditions), then the pattern claims a lot. For example, if a pattern says that it can be used for banking OR game programming OR network environments, then we have more cases in which we can actually test it. Similar to the previous example, by saying that a pattern works for network environments as well, we have eliminated the tautology "does or does not work for network environments" in favour for a decision. Note that the tautology could also be eliminated by claiming the opposite, e.g. to state in the context that the pattern is not suitable for network environments.

Generally, each situation that is not explicitly included or excluded in the context description remains unspecified – that is, the pattern makes no claims about whether or not it works in such cases. If $\{c_1, c_2, \ldots c_n\}$ is the set of all possible contexts 1..n, then a context description that states only c_1 says nothing about $c_2, \ldots c_n$. To say a pattern works in all contexts is a shortcut for saying it works in contexts $c_1, c_2, \ldots c_n$. To say a pattern works only (exclusively) in context c_1 is to say that it works in c_1 and does not work in $c_2, \ldots c_n$.

A word of caution at this point: These examples are to show how statements affect the empirical and logical content of a pattern.

This is not a call to actually use such formal statements in the written patterns! It is only to make the reader aware in which way the contexts "banking domain", "banking domain or game programming", "all domains", or "only in banking domain, no others" differ in their claims. A pattern should only claim as much as has been tested, e.g. to say it works "only in banking domains" when there is actually no data whether or not it works in other domains is as bad as claiming that a pattern works in all contexts when there are some in which it does not.

5.3.3 The more precise 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 constrained 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 instance 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 cases, 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 precise 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. We can also extend the empirical content by explicitly saying that a training of 29 or 46 minutes will never work because such claim can be tested: running a successful online training of 46 minutes would falsify the assumption that 46 minutes are inappropriate[2].

[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.

5.3.4 The less precise 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.

5.3.5 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 forces are actually resolved by a given solution.

The volume of empirical content tells us how much there is to test and how much we can learn from a pattern. However, it does not tell us how often the pattern has actually been tested. The more a pattern claims (a broader context or very detailed statements about what works and what does not) the more tests have to be performed.

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. change 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.

This section has illustrated how the empirical content changes. For clarification we have used very simple statements of which we have assumed that they can be considered in isolation and that they are discrete. By all means, this was only for illustration. You should not start to write more formal patterns with isolated statements. Not only would this contradict the usability for the target audience. It would fragment the pattern into isolated parts reducing it to mechanistic analysis and contradict the very idea of Christopher Alexander.

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 coincidence, 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:

00110101000100010000**1001**	00101010101111101011101
110100001010100011111**1001**	11110001111101111000**1001**
110100001010110011101011	10000101101011101010001
110100001010111110001110	11110110000111110001001
11111101110110111000111	

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 causal 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 that can only capture correlations. 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. Again, the binary representation was just for illustration: real design problems are far more complex and binary representations might be possible in principle but not practically.

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 [32, 33, 34, 35] were mined using the artifactual method by investigating various multimedia applications [31]. A first recurrence of interaction forms qualified the form as a pattern candidate. Another

[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 by chance than of simple patterns.

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

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, to evaluate the quality of a pattern it is of interest 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, as they provide evidence or counter-evidence for the reliability of a pattern.

6. CONCLUSION: 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 [47]), 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 application of a rule is pointed out by Aristotle already [45]. Knowing the rules of painting does not imply that you can actually paint. Tacit knowledge is always richer than any explication of that knowledge. There are things that cannot be communicated appropriately because such know-how is not only deeply in the individual but may actually depend on the individual [40]. Explicit rules are helpful to communicate good practice but it is not said that such rules actually exist internally. When we perform or solve a design problem we are not aware of our thinking processes. The reasoning comes afterwards and has the same structure as the explicit rules that we can communicate. This, however, is not a proof that we have internally the same rule-based knowledge [38].

A pattern is neither a necessary nor a sufficient condition for successful problem solving. One can solve problems without prior knowledge of patterns. And even the best pattern will not help if an individual fails to understand and internalize its meaning and successful practical application. The reliability of a pattern is a necessary condition to be useful but it is not sufficient. The quality of a pattern highly depends on its usefulness. Usefulness depends on the complexity and relevance of a problem and the communication of the pattern.

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. decides 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)

To be sceptical about each individual pattern does not weaken but strengthen it. If a pattern can fail in principle but shows to succeed continuously, this evidence makes it more reliable.

It is our goal to link the epistemic thoughts of this paper with the actual practices of pattern mining and hope to capture a pattern language for pattern mining over the next years. We welcome every critical feedback.

7. 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. Thank you, Linda! We would also like to thank all participants of the "People" Writer's Workshop group at the PLoP 2009 in Chicago, Illinois. Their encouraging and critical remarks have shaped the current version of this paper. Furthermore, thanks to everybody who joined the "Is that true…?" discussion at PLoP 2009. The comments and thoughts have been very encouraging and helpful.

8. REFERENCES

[1] Alexander, C. (1964). Notes on the synthesis of form. Cambridge: Harvard University Press.

[2] Alexander, C., Ishikawa, S., and Silverstein, M. (1968). A pattern language which generates multi-service centers. University of California, Berkely: Center for environmental structure.

[3] Alexander, C. (1979). The Timeless Way of Building. New York: Oxford University Press.

[4] Alexander, C., Ishikawa, S., and Silverstein, M. (1977). A pattern language: towns, buildings, construction. NewYork: Oxford University Press.

[5] Alexander, C. (2002a). The nature of order, Book 1. The phenomenon of life. Berkeley, Calif: Center for Environmental Structure.

[6] Alexander, C. (2002b). The nature of order, Book 2. The phenomenon of life. Berkeley, Calif: Center for Environmental Structure.

[7] Appelton, B. (2000) Patterns and Software: Essential Concepts and Terminology. http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html. (accessed July, 2009)

[8] Berkun., S. (2007). The myths of innovation. Sebastopol, CA: O'Reilly.

[9] Booch, G. (1998). Patterns. In Rising, L. The Pattern Handbook. Cambridge: Cambridge University Press.

[10] Bortz, J., & Do ring, N. (2002). Forschungsmethoden und Evaluation: Fu r Human- und Sozialwissenschaftler. Berlin [u.a.]: Springer.

[11] 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.

[12] Coplien, J. O. (1996). Software Patterns. New York: SIGS Books.

[13] Coplien, J.O. (1998). Setting the Stage. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.

[14] Coplien, J.O. (1998b). Space: The Final Frontier. C++ Report 10(3), pp. 11-17. March 1998

[15] Corfman, R. (1998). An Overview of Patterns. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.

[16] DeLano, D. E. (1998). Patterns Mining. In Rising, L. (1998). The Pattern Handbook (pp.87-96). Cambridge: Cambridge University Press.

[17] Feynman, R. (1974): Cargo Cult Science. Engineering and Science 37:7 (June 1974), p. 10–13.

[18] Flick, U. (1998). An introduction to qualitative research. London: Sage.

[19] Foote, B. (1997) In Martin, R.C., Riehle, D. & Buschmann (eds.). Pattern Languages of Program Design 3. Addisson Wesley.

[20] Gabriel, R. P. (2002). Writers workshops and the work of making things: Patterns, poetry. Boston [Mass.]: Addison-Wesley.

[21] Gabriel, R. P. (2008). Writers' Workshops As Scientific Methodology. ht tp://dreamsongs.com/Essays.html (accessed August, 2008)

[22] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley.

[23] Goldstein, E. Bruce. (2009). Sensation and Perception. Wadsworth Pub.

[24] Hanson, N.R. (1958). Patterns of Discovery. Cambridge.

[25] 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.

[26] Harrison, N.B. (2006). The Language of Shepherding. In Manolescu, D., Völter, M., and

[27] Noble, J. (Eds.). Pattern Languages of Program Design 5. Boston: Addison-Wesley.

[28] Hume, D., & Buckle, S. (2007). An enquiry concerning human understanding and other writings. Cambridge texts in the history of philosophy. Cambridge: Cambridge University Press.

[29] Kerth, N.L., Cunningham, W. (1997): Using Patterns to Improve Our Architectural Vision. IEEE Software 14(1), Seite 53-59.

[30] Kohls, C., & Scheiter, K. (2008). Mental acquisition of design patterns.. Proceedings of the 2008 conference on Pattern Languages of Programs (PLoP). Nashville, Tennessee: ACM.

[31] Kohls, C., & Uttecht, J. G. (2009). Lessons learnt in mining and writing design patterns for educational interactive graphics. Computers in Human Behavior, 25 (5), 1040-1055.

[32] Kohls, C., & Windbrake, T. (2007). Moving objects. http://hillside.net/europlop/europlop2007/workshops/F4.pdf.

[33] Kohls, C., & Windbrake, T. (2006). Towards a Pattern Language for Interactive Information Graphics. http://hillside.net/plop/2006/accepted_papers.htm.

[34] 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.

[35] 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.

[36] Majetschak, S. (2007). Ästhetik zur Einfu hrung. Hamburg: Junius.

[37] Mor, Y. & Winters, N. (2007): Design approaches in technology enhanced learning. Interactive Learning Environments, 15(1): 61-75.

[38] Neuweg, G. H. (2001). Könnerschaft und implizites Wissen: Zur lehr-lerntheoretischen Bedeutung der Erkenntnis- und Wissenstheorie Michael Polanyis. Internationale Hochschulschriften, Bd. 311. Mu nster: Waxmann.

[39] Noble, J. (1998). Classifying relationships between object-oriented design patters. Australian Software

[40] Polanyi, M. (1958). Personal Knowledge: Towards a Post-Critical Philosophy. Chicago: University Of Chicago Press.

[41] Popper, K. R. (1972). The logic of scientific discovery. London: Hutchinson.

[42] Rising, L. (1998). Design Patterns: Elements of Reusable Architectures. In Rising, L. (1998). The Pattern Handbook (pp.87-96).Cambridge: Cambridge University Press.

[43] Rising, L. (2007). Understanding the Power of Abstraction in Patterns. IEEE Software. July/August 2007.

[44] Schmolitzky, A., & Schümmer, T. (2008). Patterns for Supervising Thesis Projects. European Conference on Pattern Languages of Programs (EuroPLoP) 2008. Irsee Monastery.

[45] Schnädelbach, H. (2002). Erkenntnistheorie zur Einfu hrung. Zur Einfu hrung, 268. Hamburg: Junius.

[46] Schümmer, T., & Lukosch, S. (2007). Patterns for computer-mediated interaction. Wiley series in software design patterns. Chichester, England: John Wiley & Sons.

[47] Simon, H. A. (1969). The science of the artificial. Karl Taylor Compton lectures, 1968. Cambridge: M.I.T. Press.

[48] Simon, H.A. (1973). Does Scientific Discovery Have A Logic? Philosophy of Science. 40. p. 471- 480

[49] Taleb, N. (2007). The black swan: The impact of the highly improbable. New York: Random House.

[50] Vlissides, J. (1997). Patterns: The Top Ten Misconceptions. Object Magazine. March Issue.

[51] Wertheimer, M. (1938). Laws of organization in perceptual forms. London: Harcourt, Brace, and Jovanovitch.

[52] Watzlawick, P. (1976). How real is real?: Confusion, disinformation, communication. New York: Random House.

[53] Wertheimer, M. (1938). Laws of organization in perceptual forms. London: Harcourt, Brace, and Jovanovitch.

[54] 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.

[55] Westermann, R. (2000). Wissenschaftstheorie und Experimentalmethodik: Ein Lehrbuch zur psychologischen Methodenlehre. Go ttingen [u.a.]: Hogrefe, Verl. fu r Psychologie.

[56] Wittgenstein, L (1922). Tractacus Logico-Philosophico. London: Kegan Paul, Trench, Trubner & Co.

[57] Wittgenstein, L. (1953) Philosophische Untersuchungen, Suhrkamp (2008).

[58] Whitehead, A. N., & Russell, B. (1925). Principia mathematica. Cambridge [Eng.]: The University Press