APPLYING AND DEVELOPING PATTERNS IN TEACHING

Jens Bennedsen[1] and Ole Eriksen[2]

Abstract *- A community of teachers and researchers within computer science has adopted the idea of patterns and developed a set of pedagogical patterns. These patterns capture best practices in teaching. From our research and teaching practice we have observed that pedagogical patterns are useful, but there is a need for concepts and tools to justify and analyse the patterns.*

In every teaching community there exists a set of values characterizing what good teaching is about. Patterns may be measured by stating which values they imply and to what degree. Considering them as value based patterns in teaching will enrich the notion of pedagogical patterns.

Inspired by conditions for learning we identify three values in teaching in the field of engineering-related educations. Further we present a value-based template for guidelines in teaching, causing a better understanding of the patterns and help teachers to develop, apply and communicate patterns.

Index Terms – Pedagogical patterns, teaching programming, constructive learning, teaching values.

INTRODUCTION

Christopher Alexander says, "*Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use the solution a million times over, without ever doing it the same way twice*"[1]. Even though Alexander was talking about patterns in buildings and towns, what he says might be true about teaching. The notion of patterns is adopted with great success in object-oriented software design [8] as well as in process-oriented fields such as organization [7] and project management [5]. Patterns in the areas mentioned concern abstraction, reuse and communication. It is a highly stimulating challenge trying to extract a general solution, from a concrete successful work, in order to reuse the solution. Furthermore, there seems to be a great variation in how solutions are described, in order to communicate them. The patterns in object-oriented software development, organization and project management are described in very different ways; it seems to be the case that the pattern language is of great influence of the area.

The purpose of our research, which is documented in this article, is therefore to identify conditions that have to exist before patterns can be used in teaching as well as defining a template for pedagogical patterns stimulating teachers to develop, apply and communicate patterns effectively.

PEDAGOGICAL PATTERNS

In the following we briefly introduce the concept of pedagogical patterns and the current state of the pedagogical patterns project. The idea of the pedagogical pattern project is to write, in a uniform way, solutions to common problems in teaching object oriented programming. The pedagogical pattern project can be found at [2].

Definition of pedagogical patterns

In [2] the concept pedagogical pattern is defined as follows:

"*Patterns are designed to capture best practice in a specific domain. Pedagogical patterns try to capture expert knowledge of the practice of teaching and learning. The intent is to capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge.*

In essence a pattern solves a problem. This problem should be one that recurs in different contexts. In teaching we have many problems such as motivating students, choosing and sequencing materials, evaluating students, and the like..."

The description of patterns is based on a template, so the reader of the patterns is familiar with the form and easily finds the specific information needed. Therefore it is important; that the template is chosen with care and all aspects of the concept is taken into consideration. In [2] the template contains, among other the following elements:

- PROBLEM / ISSUE: problem, challenge, or issue that the pattern is addressing
- AUDIENCE / CONTEXT: For what type of learners, in what context, is this pattern appropriate?
- FORCES: What makes the problem a problem?
- SOLUTION: the solution this pattern proposes to the problem
- DISCUSSION: resulting content/consequences and implementation issues

Below two pedagogical patterns are extracted from the original description from [3]. They are described by some of the above-mentioned elements from the template. After a discussion of the foundation of pedagogical patterns in the

[1] Department of Computer Science, University of Aarhus, Denmark, jbb@daimi.au.dk
[2] Department of Computer Science and Engineering, Aalborg University Esbjerg, Denmark, oe@cs.aue.auc.dk

next sections, we will illustrate our points by rewriting these two pedagogical patterns.

Lay of the Land

Students are given some early experience in examining a large artefact, beyond their ability to produce, with the intent of showing them the complexity of the field they are about to study.

PROBLEM / ISSUE: Often we teach courses that cover a lot of ground. If students don't see the big picture fairly early, they may never see it while lost in a sea of detail. We would like to show students the breadth of a large topic so they have something to relate to and don't get lost in the details as the course progresses.

AUDIENCE / CONTEXT: Lay of the Land has very wide applicability to almost every domain. It is especially useful in teaching topics with a lot of parts that must fit together in certain ways. Teaching programming is one example.

FORCES: Students need to see the big picture too, as well as the detail.

Early on, they can produce only simple artefacts, but they can examine, if only in a superficial way, a complex artefact. Most people can read and understand something much more complex than they can themselves produce.

Seeing the big picture can give them motivation for the study of the parts as they have an idea of how they might be used.

SOLUTION: Give students a large artefact to examine early in the course. They can see what it is that they are supposed to be about in that course and what kinds of things they will be expected to master.

The artefact should have the complexity of something you would like them to be able to produce at the end. Spend time examining the parts and their interactions.

Fill in the blanks

Students can often learn a complex topic by building several small parts of a larger artifact. This aids both their reading and writing skills.

PROBLEM / ISSUE: Beginning students need to work on larger projects than has been typical in the past, yet they are unsophisticated and have only a little knowledge and skill at the beginning. On the other hand, they can learn by reading as well as by doing. How can you get the students working on larger artifacts without overwhelming them?

AUDIENCE / CONTEXT: This pattern is intended for programming courses that attempt to move students quickly to difficult material.

FORCES: Students should see how the work that they do fits into a larger context.

Students can learn to read programs earlier than they can learn to write them. But, they should not be permitted to be overly passive in their reading.

SOLUTION: Prepare a very well designed program or part of a program and remove a few pieces of the code. Give the result to students with instructions to fill in the missing parts.

DISCUSSION: The overall design of the artifact must be excellent. The missing pieces should be carefully selected so that students can deduce something about the missing parts from the supplied parts.

Current state of the Pedagogical Patterns Project

The pedagogical pattern project started in 1996 [12]. The pedagogical patterns have not gained much influence in the teaching community. To our knowledge there are no reports on the awareness of pedagogical patterns among teachers, no systematic evaluation of the effect of pedagogical patterns or other ways of evaluating pedagogical patterns.

It remains to be seen whether it is possible to agree upon a common template. We see this as a first necessary step in order to get a wider acceptance of the pedagogical patterns. As Fincher [9] notices, one of the problems with the work of the pedagogical patterns community is the lack of a template to describe the patterns – only one member of the community uses the template developed within the community.

At the moment the work within the community is focused on creating pattern languages for specific topics, like a pattern language for seminars [10]

PRODUCT VS. PROCESS PATTERNS

The community that developed the notion of pedagogical patterns has done a lot of research and teaching in design patterns for software construction. The members of the community have seen the power of patterns in the field of creating software with high internal quality using a minimal effort, because use of patterns implies reuse on a high level of abstraction. Design patterns primarily focus on the creation of artefacts. They focus on good ways of creating artefacts and on the properties they possess. Teaching is not about creating artefacts but about creating good learning processes. We believe that it is of great importance to take into account that teachers have to define, create, control and evaluate learning processes, so this should be the main focus for the patterns that we use for inspiration.

Design patterns have great impact on modern software development. The reason for this success might be that in design patterns and other pattern languages an underlying set of values (low coupling, high cohesion, simplicity, reusability, generality, etc.) is accepted by almost all users of the patterns. This separation of concerns (what are the values and how do we achieve them) is important because a community needs to agree on such values before or in coherence with developing a common set of principles or guidelines for practice achieving the values. We believe that one of the main reasons for the lack of success of the pedagogical patterns is the absence of generally accepted values in teaching.

The cultural background is not taken into account in the pedagogical patterns. Different countries have different traditions of teaching. This is seen in many different ways –

for example in the way students are graded. In the American educational culture it is common that each student is graded individually and during the course, as opposed to Denmark, where the student more commonly is graded at the end of the course. Also, many times the grading is based on a group project.

This implicit educational cultural background becomes visible in the pedagogical pattern "Grade it again, Sam", where the thumbnail of the pattern says: "*To provide an environment in which students can safely make errors and learn from them, permit them to resubmit previous assignments for reassessment and an improved grade*." Here the assumption is, that students submit assignments for grading, something which is very seldom done in Danish universities. As Fincher [9] notes "*The majority of the patterns appear to be codifications of single pieces of practice, or practice from single practitioners". At the current time, there seems to be little variety of input from which selections of practice, exemplifying the desired qualities, can be selected and captured*."

We believe that the pedagogical patterns can have a good impact on the teaching, but it is important to take the inherent characteristic of teaching into account when designing the patterns.

TEACHING AND LEARNING

Above we have turned the foundation of the pedagogical patterns into a problem – not the notion of them. In order to give a foundation for pedagogical patterns we need to present our understanding of the two most important concepts in this field: Learning and Teaching.

The concept of learning has many faces and there is no universal agreement on what it really covers. If we ask a couple of teachers about the purpose of their teaching they agree it is to make their students learn; but most likely they will disagree on the meaning of the word learn. In the following we present our definition of learning and discuss its relation to teaching.

Bloom [4] defines a six-layered hierarchy of knowledge and competence: understand, knowledge, application, analyse, synthesis and evaluation. We restrict our understanding of learning to a process that achieves at least the third level in Bloom's taxonomy; *thus, learning is the process of developing one's skills in creating artefacts* – generally speaking. Within the field of engineering-related educations, this is no limitation. Such educations have the overall purpose of developing the constructive skills of their students. A person learning something in a programming course increases his or her skills as a programmer. After a period of learning, the person is able to construct programs that he or she could not make before. It might involve new techniques, the use of new language facilities, etc. With this definition of learning we may turn our interest to teaching: teaching takes place in many situations and covers activities with the overall purpose that someone has to learn.

In the following we will consider teaching as a formal and planned activity aiming at the development of constructive skills. Teaching in traditional practice covers lectures, tutorials, project work etc. We consider teaching as a professional work – included in a wide range of jobs e.g. from the lecturer at the university to a role in a system development project – they both practice teaching but in different contexts.

An interesting dilemma presents itself through the distinction between teaching and learning: one might learn many useful things without any teaching taking place, and a student may be involved in lots of teaching without learning anything. This may sound a bit disappointing. The worst part of the point is that students may be involved in a lot of teaching without learning anything, and professional teachers are therefore looking for guidelines, principles, tricks etc. that may help them in their practice such that their teaching efforts cause learning to take place. A proposal of such guidelines could be pedagogical patterns. A pattern is therefore a solution. But *what are the problems*? If you ask different teachers this question you will get different answers – and most likely you will get very specific answers.

We therefore consider the situation in the following way: Before hastily focusing on solutions we identify a set of values in teaching. These values are culturally determined and they characterize the process of learning in the sense that if the values are present in a process the involved persons may learn something. Then we can answer the question asked above "What is the problem?" simply by stating: "The absence of values".

We wish to make it clear that we don't believe a universal valid set of values in teaching exist– as is the case in the field of software development. The values depend on the level of competence and knowledge you want your students to posses afterwards, the field of study, the culture etc.

Our advice is therefore that teaching communities should develop their own values, describe them in a proper way and develop or search for their own supporting patterns. In the next section we present our values and give examples of how pedagogical patterns may support them.

VALUES IN TEACHING

Above we defined learning as a process leading to the development of constructive skills. This definition does not imply guidelines for teaching, as it only describes the cognitive result of the learning process.

An old Chinese probe says that "Tell me – and I'll forget. Show me and I'll remember. Let me do it – and I'll learn it." By following this advice, teaching might be practiced by just let the students work. If, on the very first day in university, we ask a student to write a large program in an unknown programming language, in most cases nothing positive or constructive will come of it because of the built-in impossibility. On the other hand: if we ask a

senior student to solve a simple programming exercise program in a well known programming language we get a nice solution – but the activity or process that lead to the program did not result in any learning for the senior student; because it was a routine exercise.

The question addressed here is the following: There is more than just work in a process that reflects learning – what is this "more"? In [11] the author presents a simple answer to that question: *"You learn something if and only if you work with something of your interest you are almost able to finish up*". This is a very simple answer to a very complex question and therefore we have to go deeper into understanding the answer and its consequences. In [11], three simple conditions arise: You will learn something if – and only if - you

1. create[3] something in a process
2. which you are emotional involved in, and
3. this process requires skills that you almost meet.

In fact, the answer includes two messages: the first one is the "if" part. In [11] both strong arguments and experiments justify the "if" part.

The "only if" part is in some sense stronger: it expresses that there is no other way to learn: every process of learning has/needs to contain actual work on something you are emotional involved in, and furthermore the task you work on needs to be on the border of your current skills.

The answer is stated as a formula containing three both necessary and sufficient pre-conditions for learning and it is formulated with higher priority to simplicity than to preciseness or completeness. The formula can be used in a very constructive way: teach in such a way that the three pre-conditions hold for your students.

In the following we analyse how the understanding of learning as defined above may influence our understanding of pedagogical patterns.

We will regard the conditions as values, in the same sense as the Agile Manifesto [6] presents four values in software development. In the Agile Manifesto it is values because the founders of them believe that if they are present it will lead to a better software development; both with respect to the quality of the system constructed and the process in project. The agile manifesto contains four values and a dozen of supporting principles [6].

We will adopt the notion of values and supporting principles because values are easier to communicate and distinguish from principles. Firstly, we need to formulate the above stated conditions for an effective learning process as values. Secondly, we need to regard the pedagogical patterns as principles for teaching described in a structured and systematic way. We state the following values for teaching:

1. **Working students** over listening or reading students
2. **Emotional involvement** over discipline and external motivation
3. **Students working with tasks they almost are able to solve** over students working with routine or impossible tasks.

That is, while there is value in the items on the right, we value the items on the left more.

The three values are not interchangeable. This means that all three values must be present during teaching. If just one of them is missing for a period, no learning takes place. Recall that learning means enhancement of constructive skills.

The Scope of the Values

It is important to be aware of the scope of the values. Most likely/probably, teaching will fail if it is based on the values mentioned, but carried out in a context in which the values are not held to be true or not accepted by everyone involved in the teaching.. The above-mentioned values support good learning, but in some cases the student is not interested in learning; rather, he wants the diploma. In such cases, teaching based on the values is not (regarded as) good teaching.

It is our experience that bringing the values out into the open, thus explaining the students about the organization of the teaching keeps down the frustration for the students. We also believe that the constructive nature of the engineering field becomes the values well – and vice versa.

Applying the Values

After having identified fundamental values in teaching, we turn our focus to solutions; it is guidelines – described on a proper level of abstraction in a systematic and uniform way - that will establish the values in teaching. Such guidelines are similar to patterns.

In the following two pedagogical patterns are analysed with respect to the values.

Lay of the land. The idea of this pattern is that "…students are given some early experience in examining a large artifact, beyond their ability to produce, with the intent of showing them the complexity of the field they are about to study" [2].

This pattern does not directly require any work for the students. If the artifact is provided with a number of "why" questions, chances are that it implies work. So the pattern implies neither the first nor the third value. The point concerns motivation; by demonstrating a realistic artifact the students might achieve knowledge about what they are supposed to be able to build at the end of the course. This might cause motivation – because they are aware of the use of the fields in the study. For example; if the course is on data structures and algorithms the students learn about

[3] The word create is used in broad sense including abstractions, reflections, constructions, ...

different kind of containers (list, stack, set, multiset etc..) and about how they are implemented effectively. If the students have seen a concrete application using at least some of these containers this might motivate them, as it gives a sense of the "meaning" of the data structures. So the pattern supports emotional involvement – the second value.

Fill in the blanks. The idea of this pattern is "*…to prepare a very well designed program or part of a program and remove a few pieces of the code. Give the result to students with instructions to fill in the missing parts. The missing parts need to be well specified. It is also best if the result will be put to some use immediately so that students can see the effect of their work*"[2].

This pattern supports working students. So the first value is implied by this pattern. Most students may find blanks they are almost qualified to fill in, if the difficulty of the blanks varies. Under this assumption, this pattern also implies the third value. If the program is actually something in itself (an application, a game, a kernel, etc.) then this pattern may imply emotional involvement, because the students can make small changes, test and get concrete feedback (which is one of the cornerstones in motivation). In our own teaching, we have seen students fill in the blanks and after completing the compulsory work they continue working with the program. They change the functionality, make extensions, etc, and as can be imagined, highly motivated, because now the students work on exercises given by them self. In most cases they actually defined and solved problems they were almost able to complete. Therefore the pattern with this twist also implies the second value.

Our advice is to choose patterns pulling well together, in the sense that they together imply all the values.

Reflection

A teacher using pedagogical patterns should always be aware of the underlying values. Many of the patterns can be extended or twisted such that they achieve the values even higher and more equally. Let's briefly explain this point with two examples.

In the "Fill in the blanks" pattern the teacher prepares a program with blanks. But blanks can be many things. One way to introduce blanks is by giving interfaces and not including implementation of the interfaces, but merely code that uses them. In this situation the students need to make an implementation for each interface, before they get feedback from testing the program. A better way of doing this is by making dummy or minimal implementations of the interfaces. Then the students can work with the program by constructing real implementations in an incremental way, thus getting the feeling that the program converges, which is highly motivating (thus achieving value 2). Another benefit is that the students can choose to implement the interfaces with a complexity that fits their current skills; thus achieving value 3. If the teachers express the blanks as dummy or minimal implementations of interfaces, a higher degree of motivation is achieved and the students are provided with freedom in choosing subtasks fitting their current skills.

Our second example is the Lay of the land pattern. Here we achieve a lot of extra motivation by choosing an application fun to test. It does not have to be boring in order to be relevant. At the University of Aarhus in Denmark a colleague did a first year course on object-oriented programming. Java™ was used as the language for exercises and tutorials. The application he used to lay out the land was a well-written browser – called Notscape – with about one thousand lines of source code. In this application nearly all subjects (data structures, design patterns, use of invariants etc.) from the course appear. This really caught the students. Of course Notscape lacked functionality normally found in commercial browsers. After having used Notscape in Lay of the land in the beginning of the course, the teacher used it later in the course in the Fill in Blanks pattern. Reuse is useful in many situations.

In the next section we will sum up by giving a template for describing pedagogical patterns reflecting the set of values for teaching.

VALUE BASED DEFINITION OF PEDAGOGICAL PATTERNS

The pedagogical patterns need to focus more explicitly on the basic values in teaching. When we practice teaching we may use a set of pedagogical patterns. In order to choose a proper set of patterns, we need to know its effect in terms of achieving the three values. Therefore these values must be present is the template:

- **Name**: pattern name
- **Thumbnail**: A short description of the pattern with focus on what the teacher has to do and when.
- **Work**: If the pattern implies that the students have to work, what exactly do they have to create? On what basis - and how do they test it?
- **Motivation**: If the pattern results in motivation, try to explain what kind of motivation the pattern is achieving and why.
- **Capability boarder**: Explain to what degree the pattern supports students with different skills benefiting from the pattern.
- **Twist**: Shortly describe variations of the pattern, and for each of them how it influences the three values.
- **Related patterns**: Give the patterns related to this pattern, and explain what the relation is all about.

Below this template is used in a brief description of two patterns from [2].

Name: Lay of the land
Thumbnail: Give the students a large artefact early in the course. Spend time examining the parts and their interactions.
Work: The students read and understand the structure of the artefact.
Motivation: The students see in a concrete way what they can achieve at the end of the course/curriculum. The students get an overview of the necessity of the topics in the course. The artefact must be meaningful and appealing to inspect and/or use for the students.
Capability boarder: None.
Twist: Add check questions to stimulate a deeper understanding of the artefact.
Related patterns: The artefact can be used in "fill in the blanks".

Name: Fill in the blanks
Thumbnail: Give the students a well-designed artefact to work with. Develop the artefact with a lot of blanks, i.e. parts that may be subject to changes, variations or extensions. The teacher must present the artefact carefully and be concrete and specific about the blanks. The artefact must be testable and preferable a meaningful application. It is important that each blank is as isolated as possible. The artefact must be well designed in the sense that it is designed to changes and/or extensions (object-orientation may be helpful in achieving this property).
Work: The students have to develop the artefact in an incremental way by making minor changes to the artefact between successive tests.
Motivation: The students get concrete feedback from the tests. They experience that all the work makes a difference because they observe that the application grows as a result of their efforts. After filling in a lot of blanks some students may feel a kind of ownership of the artefacts.
Capability boarder: The students may choose blanks that fit their current level of competence. Some students may define blanks themselves – either easier or more difficult than the blanks initially defined by the teacher.
Twist: Prepare several artefacts that may be put together after the students have filled some blanks. This might stimulate co-operation in the class and thus result in an community in the class based on the artefact.
Related patterns: The artefact can be the one used in Lay of the Land, or part of it.

CONCLUSION

Patterns have had a great impact in many fields. In the field of teaching pedagogical patterns exists but their influence has been minimal.

In this article we have distinguished between teaching and learning. With this distinction in mind, we have discussed the influence of design patterns on pedagogical patterns and argued that design patterns focus on artefacts whereas teaching concerns the creation of good learning processes.

Patterns are solutions to problems. These solutions aim at fulfilling some underlying values. So before turning to solutions, it is important to define the underlying values. Our advice is therefore that teaching communities should develop their own values, describe them in a proper way and develop or search for their own supporting patterns. We have argued that these values should be present in the template for guidelines in teaching. Inspired by [11] we have developed our own values and analyzed how existing pedagogical patterns achieves our values. We have developed a concrete template based on our values and given examples of pedagogical patterns described in this template. We see this incorporation of teaching values in the template as a first necessary step to expand the knowledge and application of pedagogical patterns.

REFERENCES

[1] Alexander, C. et al A Pattern Language. Oxford University Press, New York, 1977.

[2] Bergin, Joseph et al, http://www.pedagogicalpatterns.org/

[3] Bergin, Joseph, "14 pedagogical Patterns", http://csis.pace.edu/~bergin/PedPat1.3.htm

[4] Bloom, B.S. (Ed.) Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. New York: Longmans, Green, 1956

[5] Cockburn, A., Project Risk Reduction Patterns (http://members.aol.com/acockburn/riskcata/riskbook.htm).

[6] Cockburn, A, Agile Software Development, Addison Wesley, 2001

[7] Coplien, J., A Development Process Generative Pattern Languages http://www.bell-labs.com/user/cope/Patterns/Process/index.html

[8] Gamma, E et al, "Design patterns – Elements of reusable software", Addison Wesley, 1995.

[9] Fincher, S et al, "Pedagogical Patterns: Their Place in the Genre", p. 199 – 202, Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, 2002

[10] Fricke, A et al, "SEMINARS A Pedagogical Pattern Language about teaching seminars effectively", EuroPLoP 2000

[11] Larsen, S.: Den ultimative formel for effektive læreprocesser, Steen Larsens forlag, 1998 (In Danish)

[12] Sharp, H et al, "Curricular Patterns", ACM SIGPLAN Notices December 1996, p 18 – 21