Learning to teach - Learning to learn

Patterns and Pedagogy, a winning team

Jutta Eckstein, Markus Voelter

Objects in Action, jeckstein@acm.org
Mathema AG, markus.voelter@mathema.de

Abstract. Pedagogical[1] Patterns capture successful experiences of learning and teaching OT, from industry or academia, in a homogeneous, easily accessible format. The Pedagogical Patterns Project started during OOPSLA '95, made its first public appearances at ECOOP '96 and then TOOLS USA '96; the first workshop was held at OOPSLA '96. Many sessions have been held at numerous conferences in America and Europe and more then 50 teaching techniques written in pattern format have been collected. The patterns themselves have proved to be useful, and have helped educators to explore and exchange their teaching experiences.

Pedagogical Patterns help to teach more effectively, by capturing teaching experience.

Why Pedagogical Patterns are needed

Most of the trainers and educators in the IT world are not educated in pedagogy. Usually, somebody teaches more or less by accident in the beginning, and based on experience later on. It is common practice to let those people teach a particular technique, who are good at using it but without considering her teaching skills. But knowing the subject is completely different than teaching it. Knowing the subject is important, but it is by far not enough.
Of course, everybody tries to do their job the best they could, but still a lot of people struggle with the problem of how to successfully teach complex subjects such as object technology (OT). Some of the trainers/educators are aware of their own lack of pedagogical skills and try different strategies in teaching a specific subject. Doing

[1] Pedagogy is a term that refers to the "systematized learning or instruction concerning principles and methods of teaching."[10].

this, they discover successful ways of teaching the specific subject, and of teaching in general.
So there is definitely a lot of pedagogical knowledge available in the heads of some experienced trainers, but it is not easily accessible for novices. They would appreciate help in these areas. That is why they are looking for possibilities to exchange their ideas and problems with others.
Imagine somebody had found a good way of teaching a specific topic and you yourself are going to teach the same topic for the first time. Imagine furthermore that this "somebody" would not only tell you the concrete trick or technique how to teach the topic but would also abstract this trick in such way that it could be used in other similar, but different contexts. Imagine, "somebody" would write it as a pattern!
A collection of patterns like these not only forms a great repository of techniques for teaching a specific subject it also defines a vocabulary which could be shared by different educators. Some patterns, the real purely pedagogical ones, have even broader scope than the specific subject in which they were found (OT, for example). They can be used in many different training environments, because they cover well-proven teaching techniques.
But even this is not the end of the story. We now have this abstraction of one or several techniques, which are called patterns. But usually, the pattern's content is part of a larger problem space. If several patterns in a common problem space are grouped and organized in a specific way, the result is a Pattern Language, the most useful and intriguing form of patterns.
The pedagogical patterns project [5] collects pedagogical patterns and combines them to form a pattern language, which enables to:

> ***Improve the quality of the process of learning by providing teachers the information on how to solve the problems that arise when teaching and learning.***

Patterns Background

The idea behind patterns is to capture expert knowledge in an accessible and usable way. The first set of patterns appeared as part of a book called "A Pattern Language – Towns, Buildings, Constructions" and was published in 1977 by the architect Christopher Alexander et.al. [1]. He defines a pattern as follows:

> **Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.**

This definition shows that a pattern is not a recipe or a kind of step-by-step documentation. Rather, it is a solution represented in an abstract, rather than a

concrete way, with the context of the problem, the forces that create the problem and the reasons for and consequences of using the solution.
Patterns are neither simple tricks nor magic techniques, though many start with just a good idea. To become a pattern, however, the idea must go through an abstraction process, so that the essence is factored out into a form that is both generally applicable, while still being practically usable. For experts in the field patterns are neither novel nor surprising, but they brings together in one place expert knowledge that is often forgotten and sometimes overlooked.
There are many symptoms that identify something rather as a technique than a pattern:

- ?? It can be used in exactly one situation
- ?? It can be done in only one specific way
- ?? It has not necessarily been proven in practice

The essence of a pattern is that the situation to which it applies recurs in different circumstances and in each context there is a set of forces that must be considered to find an appropriate solution. Patterns attempt to capture the essence of this process by stating the appropriate contexts and the forces that must be considered. Finally they present a solution to the problem in that context.
Ideally, each pattern is simple and well-focused on a specific problem. The real power of a pattern becomes visible if they are embedded in a pattern languages of related patterns in the attempt to generate emergent behavior from simple parts. The effort to create pattern languages is non-trivial and usually requires the work of many minds.
Patterns have been discovered in many areas in computer science since about 1993. Design patterns and architectural patterns are well known in the industry. Less well known, perhaps, is the work on organizational patterns, which has had great success in many industry sectors, e.g. telecommunications. Almost all of these pattern areas use a different kind of format, for supporting their contents the best. Although nowadays there is a tendency of regarding the Alexandrian – or literal – form as the most valuable one. The pedagogical patterns project follows this tendency.

Patterns for Pedagogy

There has been an international effort in writing and collecting Pedagogical Patterns with many authors contributing patterns and pattern languages, called the Pedagogical Patterns Project [5]. It started in the OT community for two main resons: First, there was the need to document successful teaching techniques, because the concepts are complex, and many trainers have no or little pedagogical skills. Second, the patterns approach was beginning to be widely used to document technical topics in the OT world. It seemed and proved to be a good way to document pedagogical knowledge. Today, people interested in pedagogical patterns still belong mostly to the OT community, but there are tendencies to widen the project to all kinds of teachers, educators and trainers [2]. For example if you take a look at one of the most mature pedagogical pattern languages – Seminars [8], you will discover, that this was

developed without OT as the basis. However, most of the other available pedagogical patterns focus particularly on OT.
Generally there is no need to limit the scope of the pedagogical patterns project to OT only, but as long as almost all contributors have OT as a background we have hardly any known use in any other area. Although we believe that the pedagogical patterns could also apply in different – none OT specific – areas.
Pedagogical patterns attempt to capture the knowledge of experienced teachers in a way that can be used (or applied) by novice teachers. They can also capture expert knowledge of teaching techniques in a certain domain. This can be valuable to experienced teachers who are unfamiliar with this specific domain, but have to teach it.

An Example Pattern

Pedagogical patterns have been written for several areas in education, among them topics like planning a seminar and then running it [8], teaching techniques and motivation of participants [4], creating exams and course material, developing a course in Computer Science [3] (which is actually still work in progress), as well as how to make good use of examples and exercises.
Below you find an example pattern called Digestible Packets. The pattern is one of 48 patterns, which together form the SEMINARS [8] pattern language. The problem space covered in SEMINARS is how to successfully run a seminar. The language covers preparation, room arrangement, social aspects, exams and many other topics.
Each Pattern starts with the description of a problem:

> ***People can only concentrate for a limited period of time. This is the primary reason to include regular breaks. If a topic takes longer than the time people can concentrate, the participants will have difficulties understanding the topic in its entirety. Because comprehension decreases, the motivation will decrease, too, and the seminar will be considered difficult.***

The problem helps you to find out whether the pattern could be important for your situation, or not. Then, there is a practically proven way to solve the problem, usually starting with the word "therefore".

> ***Therefore, organize the topics in such a way, that the topics remain small and understandable. Ideally, each topic should be understandable on its own and should be finished during a reasonable amount of time, ideally the time the participants are able to concentrate. If the topic takes longer, make a break! Be sure to review after breaks to bring them back in line. Create logically consistent packets and show the interrelations among the packets.***

For clarification and to ease the transfer "into the real world", examples are provided:

> ***Be sure that at the end of each session (or at least of each day) the topic is finished, usable, and complete. The participants will leave with a feeling of accomplishment and satisfaction. You can use a summary to augment this feeling. As a consequence, you should also avoid organizing a seminar that only covers "dry, unimportant" theory for the first three days, until at the fourth day, practically usable topics are taught. After at most two days, the participants will be disappointed and frustrated.***

Pedagogical Patterns should also be used to define a common vocabulary on teaching techniques – therefore each pattern must have a succinct and easily memorable name, in this case: ***Digestible Packets***

State of the project and current activities

The project has quite a lot of material. There are some complete pattern languages available, others are currently written. In addition, there is a set of about 60 so-called proto-patterns, small patterns that have not yet been verified by a patterns writers' workshop and are not embedded in a pattern language.
The people in the project are currently working on two main areas. One is the refactoring of the proto-patterns [7]. They are all on different levels of abstraction, and they have different quality. We aim to find the "core patterns" and combining them into a set of pattern languages, which will then be discussed at a patterns writers' workshop at a patterns conference. The resulting pattern language, which is still work-in-progress, focuses on experiential learning [6,9].
The second major effort is "going public", which means telling people about the project and its goals. We believe that pedagogical patterns are a great way to achieve their stated goals. The more people use these patterns or contribute new ones, the better so, join us.

How to contribute

As mentioned above, pedagogical patterns solely exist to help educators/trainers in industry and academia to do a better job. To make this a successful attempt, we need more people who help here. If you are interested in participating, you can do this in several roles.

?? **Test pilot:** Like all patterns, also the pedagogical patterns are proven and not invented, but still every pattern benefits from continuous usage. The job of a "Pattern Test pilot" is to try the patterns in reality and give feedback to the author.

- ?? **Reviewer:** The more people see and read a pattern, the better the pattern gets. This is especially true for the non-native speakers, who write the patterns in a foreign language. The job of a reviewer is to read patterns and give the author feedback on content, language and form.
- ?? **Author:** Of course, somebody has to write the patterns.
- ?? **Advertiser:** One of the big problems of the project in its current state is, that too few people know about it. It is therefore crucial, that we gain more publicity.

Any kind of participation is appreciated. If you're interested, please visit us on our homepage at www.pedagogicalpatterns.org.

Conclusions

In general, teaching is about flexibility. Every teacher should have a set of techniques to run a good seminar. A teacher should collect these effective techniques over time, perhaps creating a personal pattern language or finding new patterns to add to this pattern language.
Specific examples are indeed very valuable, but for being widely usable the essence has to be factored out so they can be shared as patterns. Combining those patterns in a pedagogical framework, should allow to improve the quality of the process of learning object technology by providing teachers the information on how to solve the problems that arise when teaching and learning object technology.

References

1. Alexander, Christopher et.al. *A Pattern Language: Towns - Buildings - Construction*. Oxford University Press. 1977.

2. Bergin, Joseph. Pedagogical Patterns and Pattern Languages for the Early CS Courses. Working Group Submission to ITISCE 2001, a SIGCSE /SIGCUE conference on Innovation and technology in computer science education. http://www.cs.ukc.ac.uk/events/iticse2001/

3. Bergin, Joseph, *Fourteen Pedagogical Patterns*, In: Proceedings of EuroPLoP 2000.

4. Eckstein, Jutta, *Learning to teach and Learning to learn*. In: Proceedings of EuroPLoP 2000.

5. Homepage of the Pedagogical Patterns Project: http://www.pedagogicalpatterns.org

6. Marquardt Klaus, Eckstein Jutta, Voelter Markus. *Experiential Learning. Part 1*. Proceedings of EuroPLoP 2001. http://www.hillside.net/patterns/EuroPLoP

7. Sharp, Helen; Manns, Mary Lynn; Eckstein, Jutta. *The Pedagogical Patterns Project*. In: OOPSLA 2000 Addendum.

8. Voelter, Markus; Fricke, Astrid. *SEMINARS, - A Pedagogical Pattern Language about teaching seminars effectively*, Proceedings of EuroPLOP 2000, or at http://www.voelter.de/seminars

9. Wallingford Eugene, Manns Mary Lynn. *Experiential Learning Part 2.* Proceedings of PloP 2001. http://jerry.cs.uiuc.edu/~plop/plop2001/

10. Webster's New Collegiate Dictionary. G&C Merriam Co., 1959.