

## **Introduction to Pedagogical Patterns**

Pedagogical patterns, like all patterns, attempt to capture expert practice. In this case it is the practice of experienced teachers, both in academia and in industrial settings. The papers in this section propose successful techniques to assist with teaching and learning. For many professional educators, these patterns may at first sight seem obvious, even trivial. However, all educators both experienced and novice will benefit from the ideas contained in these languages. For those newer to teaching pedagogical patterns offer a way to receive the wisdom passed on by experienced teachers. But even the experienced will benefit from patterns because they offer a way to learn from one another. This is, of course, because Nobody is Perfect [VF], and furthermore everybody has developed her own little secrets that she can share.

The pedagogical patterns project [PPP] is working on collecting many types of patterns that can help teachers teach and students learn. This collection demonstrates recent efforts by members of the project.

These patterns, based on earlier work, were revised and rewritten in Alexandrian form in order to support the integration into a pattern language. The work continues and additional patterns will be submitted to future conferences of the PLoP series.

## **The Pedagogical Patterns Project**

Most educators and trainers are not taught how to teach. Rather, they often find themselves teaching by accident. Typically, a person with a skill that is in demand, such as a particular programming language, will be asked to teach it. People assume that if the person is good in this programming language, she will be good at teaching it. But knowing the subject matter is very different from knowing how to teach it. Moreover, the successful teaching techniques of non-technical fields may not be known to many instructors of technical subjects.

Effectively communicating complex technologies is often a struggle for information technology instructors. They may try various teaching strategies, but this trial and error process can be time-consuming and fraught with error. Advice is often sought from other “expert” instructors, but these individuals are not always readily available, nor do they often reflect on why they are successful. This creates the need to find other ways to facilitate the sharing of teaching techniques between expert and novice teachers.

This is the goal of the Pedagogical Patterns Project (PPP, [PPP]). Pedagogy is a term that refers to the “systematized learning or instruction concerning principles and methods of teaching” [Web]. Patterns provide a method for capturing and communicating knowledge such as pedagogy. As an example, imagine that you are looking for an effective way to teach message passing to experienced programmers in a weeklong industry course. A friend who is

teaching a semester-long object technology course to traditional age university students has found an effective technique. He shares it with you without dictating the specific implementation details. This allows you to use your own creativity to implement the technique in a way that is most comfortable for you and most useful for your industry students. This is the essence of patterns – to offer a format and a process for sharing successful practices in a way that allows each practice to be used by a variety of people in many different ways.

A collection of patterns could form a repository of techniques for teaching a specific subject, such as object technology (OT), or for covering a particular aspect of pedagogy (as in these languages). Ideally, many of the patterns would have an even broader scope than OT, but all of them would be useful in many different training or learning environments because they are proven teaching techniques.

But even this is not the end of the story. Related patterns can be combined in either a pattern catalog [Bus] or in a system of patterns [Fow]. A third possibility is to relate several patterns within a common problem space, the result of which is a language of patterns that provides a resource for solving complex problems. The goal of the project described in this paper is to form pedagogical pattern languages for teaching. This will provide instructors with the ability to share their effective teaching techniques in a common format, to document relationships between the techniques and to form powerful tools known as pattern languages.

The Pedagogical Patterns Project (<http://www.pedagogicalpatterns.org>) is a continuing endeavor. We invite all interested parties to join in these efforts to capture expert practice in a transferable way.

## The Pattern Language

The patterns in these pattern languages use a form similar to the one used by Alexander in his book *A Pattern Language* [CA]. All patterns are written in the you-form, thus directly talking to you, the teacher. In addition to the pattern name, each pattern is divided into four sections. The sections are separated by \*\*\*. The first section sets the context. The second describes the forces and the key problem. The third section outlines the solution, the consequences, limitations and disadvantages. The fourth section complements the discussion of the solution, by providing further information and examples. The key problem and the solution are in bold font and represent the thumbnail of the pattern (also called the pattlet). The examples are in italic font. References to patterns inside this pattern language are in CAPITAL LETTERS, references to patterns published elsewhere are in normal font, but followed with the [pointer] to the reference section.

In addition, each pattern is marked with one or two asterisks (\*), as in Alexander's patterns. They show how fundamental we believe the pattern is.

Two asterisks denote patterns that state a true invariant. We believe that it is not possible to solve the stated problem properly, without referring to the solution that we have given. One asterisk means that we think that we are on the right track, but we believe it will be possible to improve the solution.

A few of the patterns in this language have been developed earlier as part of other pattern languages. They are marked with a *DE*. Thumbnail sketches of these are placed inline here, with references to a fuller development elsewhere.

## **Annex – Editors Background**

The Pedagogical Patterns Project is an international one with individuals from more than twelve different countries. Despite the fact that both the authors of the original patterns, as well as the editors of this paper come from completely different environments with varying circumstances they are still able to agree on these patterns.

**Joseph Bergin** is professor of computer science at Pace University in New York City. His career has been spent almost entirely teaching small classes in a traditional university environment, though he now uses technology extensively in all his courses. He also teaches some hybrid courses that meet a few times and carry on extensively in cyberspace using simple communication tools. Unlike Jutta, he has had no formal training in teaching itself. Thirty years of trial and (much) error has been the necessary crucible.

**Jutta Eckstein** is an independent trainer and consultant in Braunschweig, Germany. She has developed object-oriented software since 1990 and since 1991 she has been designing and teaching OT courses in industry. Having completed a course of teacher training and led many 'train the trainer' programs in industry, her main focus is on techniques, which help teach OT.

**Mary Lynn Manns** is on the faculty at the University of North Carolina at Asheville. Her area is Management Information Systems. She has over 20 years of teaching experience in the departments of Computer Science and Management & Accountancy. In 1995, she won the Distinguished Teacher Award in the Natural Sciences.

**Helen Sharp** is currently a member of academic staff at The Open University, a large distance education establishment in the UK. In addition to writing, chairing and tutoring several distance education courses, she has also taught in other, face-to-face environments. She teaches courses mainly in software engineering, object technology and interaction design. Her research work focuses on the social aspects of software development, and how software engineers develop code in practice.

## References

- [Bus] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*. Chichester, England: John Wiley & Sons.
- [CA] Christopher Alexander et.al., *A Pattern Language: Towns – Buildings – Construction*. Oxford University Press 1977
- [Fow] Fowler, Martin (1997). *Analysis Patterns. Reusable Object Models*. Reading, MA: Addison-Wesley Longman, Inc.
- [PPP] Pedagogical Patterns Project Home: [www.pedagogicalpatterns.org](http://www.pedagogicalpatterns.org)
- [VF] Markus Voelter, Astrid Fricke, *SEMINARS*,  
<http://www.voelter.de/seminars>