

# Workshop Report on:

## The Pedagogical Patterns Project

### Successes in Teaching Object Technology

**Jutta Eckstein**

Objects in Action

Thierschstr. 20  
80538 Munich, Germany  
+49-700-JECKSTEIN  
jeckstein@acm.org

<http://jeckstein.com>

#### **ABSTRACT**

The project was initiated by Mary Lynn Manns, Maximo Prieto, Phil Mc Laughlin and Helen Sharp with the goal to collect reusable pedagogical design patterns, which provide an ability to communicate proven solutions to common problems in teaching. This project began in 1996 and was motivated by the fact that although many good pedagogical ideas were being presented at OO conferences and published in proceedings and journals each year, very little had been done to collate the effective practices of many OO educators into one publication. The purpose of the pedagogical patterns project is do just that, to create a publication which is similar to what Susan Lilly (in 1/96 Object Magazine) refers to as "reusable pedagogical design patterns".

Since the project started, we have collected several pedagogical patterns. Not only have they completely different focuses (e.g. whole course concerns or how to teach a specific topic, like accessors/mutators), they also use different pattern formats.

In this workshop we reviewed the existing patterns by drawing on our own teaching and learning experience.

#### **Format**

This workshop has been split into two sessions. The first session was more an introductory session and concentrated on reviewing the existing patterns whereas the second session focused on how to proceed in the project.

The FocusGroup (or rather Workshop) started off with a brief explanation about the background and history of the project. The participants have then been asked to write on cards, what they expect from a pedagogical pattern. We tried to cluster these expectations in different categories.

We formed teams of four, where each group got two existing pedagogical patterns to read. The task was to figure out if the pedagogical patterns meet our expectations.

In a final discussion with the whole group, it seemed that 30-50% of the reviewed pedagogical patterns do not meet the expectations!

In the second session we rephrased the expectations as questions and deleted the redundant cards.

We figured out that there are mainly two main focuses: How to teach, especially dealing with pedagogical issues and Concrete Resources for Examples and Lab.

We then started to develop a Todo List, which shall help us that the pedagogical patterns will meet our expectations in the future. Afterwards we thought about *how* we could accomplish these tasks and *who* can do that.

## **Result**

*Vision of the Pedagogical Pattern Project:*

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. This information can be provided as guidelines, best practices and patterns.

*Overview about the influencing factors:*

*Scope and Boundary*

The influencing factors will be supported by a pedagogical pattern, as soon as the pattern answers some of these questions. Basically these questions could be treated as minimal requirements to a pedagogical pattern, this way they define the scope and boundary of the project.

*What follows is a collection of the expectations formulated as questions:*

## **How to teach?**

*Pedagogical Issues*

Learning to Teach:

- How to motivate students (with students we mean participants in the course. So no conclusion if this is only a concern of industry, or of academia...) in a specific OO issue? (e.g. give an orientation - via example and motivation - to some specific OO issue)
- How could the complete lecture be set up? (e.g. spiral development explained in a spiral way and not waterfall.)
- Professional educators are educated in the foundations of teaching, but how can semi-professional educators learn how to teach?
- How can I rationalize my own teaching process?
- How do you balance hands-on experience with reflection? (e.g. learning by doing versus learning by thinking)
- Do cultural dependencies affect the way OO is learned and taught?
- How do extend old (but good) ideas to new subjects?
- How do you provide the exchange of good ideas between educators?

Sequence and Interplay:

- How to sequence OO topics from the basic issues to the more demanding ones?
- What is the ideal interplay, in relation of time, between lecturing, self studies and labs? (Consider that lectures are important for developing the basic knowhow; labs are important for experimenting and getting a feel for the problems and self-studies are important for applying the knowledge to new situations.)

### *Social Issues*

#### Learning to Learn:

- How do you support and/or create the students will to learn?
- How do you teach effective learning or rather learning to learn?

#### Real World Relation:

- How can you move forward in the project while educating? (e.g. training on the job)
- How do you show the applicability of the training contents? (e.g. combine learning and production and/or provide usable technical results)

### *OO Issues*

#### Paradigm Shift:

- How can the positive and the negative effects of the fact that people have a background in other programming paradigms be used for teaching the object-oriented paradigm?
- How can OO be taught on top of existing valuable knowledge?
- How to replace "function-thinking" without deleting experience?

#### Basic Concepts:

- How to present the basic concepts, like encapsulation, inheritance, good style, tests, ...?
- What is the most important thing to teach in OO?
- How can you help people to understand the OO paradigm?
- How do you teach abstraction, esp. undergraduates?

#### Patterns:

- How can you find the appropriate pattern in a large number of patterns?
- How can you modify the pattern which is the one that fits the best to your problem?
- What is the appropriate amount of contextual information that should be supplied with a pattern, so students can understand and use it?
- What are the requirements, so students can understand the concepts of patterns? -> When to teach patterns?

### **What to teach?**

#### *Development Process*

Real World Relation:

- How to capture the essentials of the software development process in a training environment?

Team Issues:

- How can an existing team become productive?
- How can we teach team related aspects, like team design, team work, team roles?
- How can we integrate new-comers into a team?

Software Evolution:

- How can we teach software evolution and how can we deal with the related problems?

*OO Features*

Real World Relation:

- What must a manager know or learn about OO? And on the other hand what's unimportant for a manager?
- How can OO techniques be transferred into large projects?

Quality Issues:

- How to teach the effect of overdose or abuse?
- How do you teach design quality and design heuristics?
- How can you ensure that people really understood what they've been taught? How do you detect misinterpretations?

Dependency Management:

- When and how do you teach dependency management between classes and components?

**Which resources to use?**

*General*

Example Categories - What kinds of examples should be selected?

- Which topic? - Is the topic relevant for the students? Do students understand the domain problem?
- Should they be abstract or more practical? (e.g. the example could be in such a form so it can be reused later by the students)
- Is it better to use several small examples (one per topic) or a running example?

References

- Where do I find references to other teaching material?

Historical Evolution:

- How do you explain the historical evolution of a topic? But also: Is it necessary to explain the evolution of a topic? (e.g. The implementation of inheritance in programming languages)

*Concrete Examples*

Analogies:

- Where do I find an analogous description from the "real world" (non computer example)?
- How do I identify and discover analogies that work?

Known Uses:

- How could the topic be best explained? (e.g. animation)
- Where do I find known uses of the topic?

*Suggestions for Lab*

Tools:

- Which kind of tools support the education goal?
- How much tool support should be used at all?
- Are there tools which suite extraordinary good for teaching?
- Do we have to teach with the tools, the students will use later on, or should we focus on the basic commonalities between the tools?
- What are the requirements for a tool, which is used in a training environment?

Languages:

- How much idiomatic information should be included when teaching an OOPL as a first language?
- Which language is ideal for teaching OT and what requirements must the language fulfill?
- Should objects be taught first?

Physical Resources:

- Where can we find cheap/free resources that students can use effectively?
- What are the minimum resources (hardware, software, documentation, courses) to teach OO?

#### **Future Steps**

Evaluate existing pedagogical patterns by using the radar diagram, which could be found in the appendix.

Write better patterns with the formulated expectations about a pedagogical pattern as the background.

Spread the word by publishing articles, using the web and most important - publish a book.

Integrate experts from other domains ("real" pedagogical experts), e.g. for improving the quality of our patterns.

#### **ACKNOWLEDGMENTS**

I would like to thank all the participants in the Focus Group at the EuroPLoP 1999 in Kloster Irsee, Germany for their valuable contributions. Special thanks to Ademar Aguiar, who formulated the vision for the project.

#### **REFERENCES**

1. Lilly, Susan. Patterns for pedagogy. In *Object Magazine* 5(8), Jan. 1996, 96-95.
2. Mary Lynn Manns, Maximo Prieto, Phil Mc Laughlin and Helen Sharp: The Pedagogical patterns Project: Successes in Teaching Object Technology. Url: <http://www-lifia.info.unlp.edu.ar/ppp/>

3. Manns, Mary Lynn; Sharp, Helen; McLaughlin, Phil and Prieto, Maximo. *Capturing Successful Practices in OT Education and Training*. In *Journal of Object Oriented Programming* Vol. 11, No. 1, March/April 1998, 29-34.

**Appendix:**

A radar diagram that could be used to verify the existing patterns and how they meet the objectives of the pedagogical patterns project

