What Makes A Good Pattern Language?

Andreas Rüping

sd&m AG
software design & management
Lübecker Straße 1, 22087 Hamburg, Germany
rueping@acm.org

Introduction

The following is a collection of ideas associated with the question, what makes a good pattern language? These ideas fall into two categories. The first addresses the question, how can a pattern language be mined? How can we identify the patterns we would like to describe? The second question deals with how the pattern language can be shaped, and therefore mostly addresses the language quality. In other words, what criteria should be fulfilled that allow us to speak of a pattern language, rather than a mere collection of patterns.

The ideas are just that — ideas. They are not to be regarded as proven concepts. Some of these ideas I'm quite convinced of, others are rather vague. Their intended use is for a focus group on the quality of pattern languages at EuroPLoP 2002.

1 How To Mine A Pattern Language

- Mining a pattern language requires experience not just with an individual technique, but rather with an entire topic, an area, a domain. Make sure you're looking at this topic in its entirety. Look for a set of related processes or techniques, and the constraints and dependencies between them. Observing forces and counter-forces probably helps understand the domain thoroughly.
- Pattern languages should be specific. There is hardly a point writing a pattern language which consists of slight variations of existing patterns. Pattern langauge should mine patterns from a self-contained topic and a well-defined context.
- Talk to other people. Attend project reflection meetings. Attend workshops. A pattern language is rarely based on the experiences of one individual person, so experience exchange with others is a good help.
- Mining a pattern language requires a mood for reflection.You have to distill mechanisms or processes that occur repeatedly in things you observe. A pencil and pieces of paper are probably all you need to take notes and to set up a tentative roadmap diagram that shows the dependencies between the patterns you found.

2 How To Shape A Pattern Language

- A pattern language is alive to the extent its patterns relate to each other. The most common relationship is given by the pattern context — when one pattern provides the context for another. But other kinds of relationship are possible; patterns can complement each other, represent alternative solutions, etc. Ideally, the patterns permeate each other.
- A pattern languages gets an even texture when all patterns are about the same length, and closely related to this, all patterns have the same level of granularity.
- Individual patterns should neither be trivial nor too complex. A good rule of thumb is that a good pattern has a distinct set of forces, which differs from other patterns' forces. If the set of forces can be partitioned, there is probably more than pattern there. If several patterns have similar sets of forces, it is likely that they describe variations of one solution and should rather be mapped onto one pattern.
- A roadmap diagram is a good way to start a pattern language. This diagram should include all patterns, and should briefly describe the kind of relationships that hold between them.
- Many pattern languages profit from a running example. The running example can both demonstrate the individual patterns and the relationships betwen them.
- Large pattern languages deserve a good dose of structure. The good old 7 +/- 2 rule applies here. If a pattern language includes more than, say, 10 patterns, try to organise it in sub-languages which should each be as self-contained as possible. The roadmap diagram can probably suggest how the pattern language can be organised into parts.