

What Makes Pattern Languages Work Well

Klaus Marquardt
Käthe-Kollwitz-Weg 14
D-23558 Lübeck, Germany
Email: marquardt@acm.org

The Motivation behind Patterns

The key motivation behind patterns is to make information, experience and knowledge accessible to peers that do not consciously share this knowledge. These may be experts that recognise the solutions from their own experience, so that patterns create a common vocabulary for more efficient and precise communication. Furthermore, attempts are made to transfer this knowledge to newcomers so that they apply proven solutions without gathering all the (tedious and expensive) experience themselves.

What Makes Patterns Work Well

Key to the success of patterns is the accessibility of information. The mostly formal presentation does help. The forces from the pattern audience are difficult to fulfil: Some seek for a very brief description of the idea, others for a detailed description of a ready-to-use example, preferably including several variations.

Most successful patterns resolve this conflict by combining a very abstract, customisable description with a concrete example implementation. The most common approach is to describe the idea first, within a few sections or in Alexanderian form, and give the concrete implementation in a later section or as an appendix.

From Patterns to Pattern Languages

Single patterns show some deficits that are hard to grasp from the above mentioned. With a growing body of pattern literature, it became obvious that there are numerous links between different patterns, and that patterns seldom appear in isolation. Additionally, it is easier to understand (and also to explain) individual patterns when other patterns are cross-referenced, and an entire building of mutual supportive patterns in a common context is erected. The motivation behind pattern languages is also to strive for completeness within the tackled context or topic.

Do Pattern Languages Work?

The conflict between brief ideas and concrete implementation, visible already in single patterns, becomes even more evident in pattern languages. Additionally a publishing conflict is added: While languages of detailed patterns including examples are most interesting to readers that are already experts familiar with the domain (like POSA2), those languages consisting of briefly described patterns fill maybe about 50 pages, and thus fit neither into a magazine nor into a book.

Currently, I see pattern languages in a two-fold problem space: They are inherently not accessible to both kinds of pattern audience, and they have difficulties to reach their audience at all.

What Could Make Pattern Languages Work Well

The publishing problem can be solved by embedding the pattern language into the larger context of a book that treats a specific topic, of which patterns are only a (sometimes minor) part. The inherent problem of the accessibility goes deeper and should be the main topic of this focus group.

Sequences work towards the accessibility goal. Sequences are stories, fictitious or real life, that describe a logical and timely order in which (most of) the patterns within the language could be applied. They provide a link between patterns that gives them a common meaning, and potentially fetch the reader where his own experience ends. A small number of sequences at the start of a language consisting of rather detailed patterns could make the language work.

Accessibility could be gained when a reader can find his particular problem. So far, patterns usually relate to a solution, starting with their name – allowing for quick access when a reader already has an idea of the solution, and enabling communication about different solution alternatives.

To find patterns for a particular problem at hand, sequences are a possible but inconvenient way. More convenient would be a problem index, which could look like a collection of thumbnails of all patterns within the pattern language.

My goal would be to satisfy three levels of readers' interests: A quick look whether the pattern language could be helpful at all, a second look for an overview and the relations between the included patterns, and (much later maybe) a close look at the implementation details, when the application of a particular set of patterns has been decided.

Alternatives

Driving the problem-oriented approach very far, you are about to leave the district of pattern languages, and of patterns at all. Despite of the lost coherence, let's see how this works.

The analogy to problems stems from medicine: Symptoms and diagnoses. The reader experiences the symptoms, and (at least in our domain) has an idea about the diagnoses. When solutions are matched to therapies, the art of healing is to find the most appropriate therapy for a given diagnosis within a given context.

This approach is mainly focused on problem accessibility, and always offers a variety of solutions that may or may not be appropriate for a specific reader. Another possible advantage: the diagnoses do not necessarily relate patterns to each other, but include also techniques – that are different from patterns in that they do not balance forces, but the force balancing is done in selecting the appropriate therapy. So the diagnoses play the roles of both sequences, and balancing forces.

The medical analogy also brings liabilities. The suggested therapies (patterns) can hardly be driven towards completeness (i.e. pattern languages). The diagnoses may in theory cover a domain completely, but the patterns as therapies do not cover a specific topic entirely. The coherence that pattern languages can provide is definitely lost.