

What is a Pattern Language?

Sally Fincher

University of Kent at Canterbury
Canterbury, Kent, UK
+44 1227 824061
S.A.Fincher@ukc.ac.uk

ABSTRACT

In this paper, I describe work undertaken to identify the common elements of works described as Pattern Languages, with the view to delineating elements necessary to a definition of the genre. Some implications for this in terms of a possible Pattern Language for HCI are then raised.

Keywords

Pattern, Pattern Language

RESEARCH DESCRIPTION

This analysis uses *A Pattern Language* [1] (PL), *Design Patterns* [4] (DP) and Pedagogical Patterns for Teaching Object Technology [7] (PPTOT) as examples. These were examined for common (and necessary) elements which could be taken to define an individual pattern and the genre of Pattern Languages itself.

In an analysis of the form, I identify four major elements (and one minor one) which characterise patterns and pattern languages. The major elements are: Capture of Practice, Abstraction, Organising Principle and Value System; the minor element is Presentation.

RESEARCH FINDINGS

Capture of Practice

A pattern must contain a specific example of practice, because patterns aim to convey knowledge about design of *environments* (be they architectural, software or interactional) not “design” in the abstract. However, the piece of practice, the example that demonstrates and illustrates the application of the design principle, is only one constituent. Bayle et al [2] discovered within the context of a CHI’97 workshop that it was relatively easy to observe phenomena which could be put into a pattern-like form, but this act of capture was not sufficient.

The additional requirement which turns practice into patterns is an intentional and creative process on the part of the pattern author(s). Undertaking this process necessitates active consideration of the other three major elements.

Abstraction

If patterns are really a format for capturing exemplars of *design* then it is not enough that they capture any practice, the practice captured must be illustrative of a *successful*

way to solve a given problem. The characteristics which make for success must be abstracted from the example; the “good way” is then made understandable and therefore transferable to other practitioners in other situations.

How this abstraction is achieved is difficult to observe. Alexander notes that the observation of common structures in many separate and disparate environments prompted his team to think that the commonality of their design reflected something fundamental in the way people like to live in buildings [1]. Other pattern authors also claim that abstraction is a product of observing a number of separate incarnations of a single solution, and sometimes require that no practice can be a pattern unless three examples of its use can be found (the so-called “Rule of Three”).

Abstraction also serves a second purpose, that of cohesion of ideas. Practice can be captured at any scale, but it is the combination of capture and abstraction that makes the presentation of the ideas coherent. Lakoff [6] presents an example of this coherent use of abstraction in regard to the Linnean taxonomy of botanical classification.

An oak tree (for example) can be categorised at any level. However, in folk-classification (as opposed to scientific) Lakoff [6, p.35] notes that the most commonly used (and by extension, the most significant) name and reference is at the level of abstraction that corresponds to the genus (“oak”) rather than the life-form (“tree”) or variety (“white oak”) level. Linnaeus actively used folk criteria for the genus level of abstraction, which corresponds to the most readily apprehended criteria in “the real world”.

This is a concept equally important in Object-Orientation. Booch’s [3] codification of “key abstractions”, notes that there are levels which are more significant in the problem space, and useful to the solution design, than others. He, too, suggests that these might most effectively be identified from actual usage “if the domain expert talks about it, then the abstraction is usually important”. What has been noticed here is that some categories of abstraction are more basic, more meaningful to human beings in their relationship with the world than others; that is the level of abstraction that good patterns seek to embrace.

Organising Principle

Patterns do not exist by themselves, but within a framework: a catalogue or “language”. In a catalogue, the power of the collection resides in the material collected. The index, or finding aid is simply a mechanism to get to the information. In a dictionary, encyclopaedia or thesaurus, the power resides as much in the arrangement of material, in the power that the organising principle confers, as in the

individual entries themselves. The organising principle of a pattern language has a similar gestalt power; the language captures not only the pieces of design, but the shape of the whole into which the pieces fit.

The PL organising principle is scale. It recognises the impossibility of providing a complete solution, so presents many small, transferable solutions arranged in categories of scale, from “city-relevant” to “house-relevant”. Consequently, there are several entry points to the most appropriate level of patterns. The boundaries for these categories, however, are not hard and PL provides pointers, both towards larger-scale patterns to which a given pattern is contributory and to smaller-scale patterns on which it rests. The DP framework is simpler, residing on different functionality in the design process (Creational, Structural or Behavioural).

Value System

All Pattern Languages embody values, but these are not explicated in their construction. This is analogous to IQ tests which are internally consistent, valid and predictive (as are measurements of height). Their value, however, is neither measured nor contained within the application of the test but is determined by a separate, external system. A society which values high IQ (or tall people) gives a separate – and extrinsic – meaning to the results.

In similar fashion, the value-system of pattern languages is reflected by, and embodied in, their sense of audience. PL patterns have at least two audiences: architects, and the inhabitants of the buildings. PPTOT patterns have two audiences, the teachers and the recipients of teaching. This means that there are two value systems at work - one of the designer (“professional”) and one of the recipient (“user”). DP patterns have but one audience – designers – and reflect a single system of purely professional values.

Presentation

The common part of pattern presentation, and a very strong one, is the inclusion of a concrete example of an implementation of the pattern. This is not the textual description that forms the body of the pattern. In PL, a photograph conveys this example of implementation, in DP patterns it is the code sample. I believe that the purpose of these components is to sensitise the reader to the application of the pattern.

In looking at a photograph, a reaction is invoked. The intention is that the reaction is favourable—“Wow, that’s good. I’d like to live there”—and the reader is sensitised so that the information that the rest of the pattern contains becomes more accessible, more useful. For the point of the *use* of patterns is to invoke a reaction in their audience. The patterns themselves must convey information, must be information-dense to allow the pattern-user to understand them and construct their own specifics from them. But the desired consequence of using the patterns is not the transmission of information but the invocation of a reaction

IMPLICATIONS FOR A PATTERN LANGUAGE FOR HCI

If this analysis of essential elements is correct, then *any* pattern language must contain them all. The implications for HCI are interesting. The elements of Capture of Practice and Abstraction (whilst not straightforward) are in the hands of the pattern author community. The element of Value System is *less* problematic for HCI than other fields as the consideration of values (and the intersection of distinct value systems) is explicit in the domain. The presentation element, however, may be harder. HCI solutions often include a temporal or causal dimension that is difficult to represent in the traditional pattern form. It may be that HCI patterns would be better served by a dynamic presentation format.

What might constitute an Organising Principle is an open problem, and difficult to construct. A potentially interesting approach may be one espoused by Jacobson et al [5] who describe an organising principle for design notions as the balance achieved on various axes of contrast, with “good” design representing an equilibrium along and between these scales.

FURTHER INVESTIGATION

Potential extensions of this work are, first, the examination of other (possibly less mature) pattern languages to see if this construct of definitional elements is accurate and sufficient. Second, as a tool for HCI pattern authors, to inform the discourse of pattern writing.

REFERENCES

1. Alexander, C, Ishikawa, S, Silverstein, M, *A Pattern Language: Towns, Buildings, Constructions*. OUP 1977
2. Bayle, E, et al. Putting it all Together: Towards a Pattern Language for Interaction Design. In *SIGCHI 30* (1), 1998, pp17-24
3. Booch, G. *Object-Oriented Analysis and Design* Benjamin/Cummings 1994
4. Gamma, E, Helm, R, Johnson, R, Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software* Addison-Wesley 1994
5. Jacobson, M, Silverstein, M, Winslow, B. *The Good House: Contrast as a Design Tool* The Taunton Press 1990
6. Lakoff, G, *Women, Fire and Dangerous Things: What Categories Reveal about the Mind* University of Chicago Press, 1987
7. Pedagogic Patterns: Successes in Teaching Object Technology. <http://www-lifia.info.inip.edu.ar/ppp> (last update 12th August 1998)