

“Capture of Practice”: Is it obvious?

Sally Fincher, Computing Laboratory, University of Kent, Canterbury
S.A.Fincher@ukc.ac.uk

I have previously posited that there are five elements that, taken together, uniquely distinguish Pattern Languages from other forms of knowledge transfer (such as encyclopaedias, dictionaries, style guides etc.) (Fincher, 1999a; Fincher, 1999b) . I have identified these five elements as:

- **Capture of Practice:** what kind and sort of practices (and materials) patterns draw on
- **Abstraction:** why it is necessary for patterns to present an abstraction from “real world” descriptions and why it is important to consider the level and degree of abstraction.
- **Organising Principle:** how patterns are related to each other and how to consider constructing the “language” that allows them to be presented as a coherent whole
- **Value System:** Why values are central to patterns.
- **Presentation:** how the form in which patterns are presented impacts on their use.

I have started a compilation of forms of presentation at *The Pattern Gallery* (Fincher, 2000) and the third of these features (the organising principle) was more thoroughly explored with regard to HCI at the Interact’99 patterns workshop, and reported on at the CHI’00 patterns workshop.(Fincher & Windsor, 2000). This paper is concerned with the first of these elements, *capture of practice*.

It is tempting to overlook capture of practice as being a trivial aspect, too obvious to comment on,

“a pattern must contain a specific example of practice, because patterns aim to convey knowledge about design of *environments* (be they architectural, software or pedagogic) 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 ... Capture of Practice is a necessary element, certainly, but by no means sufficient. Ultimately, it is the least part of the achievement.”(Fincher, 1999a)

It may be that the nature of this capture might not be quite so “obvious” and (perhaps especially in regard to UI design) may be a more subtle phenomenon.

Craft nature of Design

In the literature of design as a distinct activity there is a thread which considers the nature and relationship of craft and design. Some of this has directed attention at software in general and HCI in particular.(Wroblewski, 1991) (McCullough, 1998). Wroblewski even goes so far as to claim “In the construction of human-computer interfaces, a craft perspective is not only in evidence but inevitable”. The ways in which the notion of craftwork have been applied in this domain are, however, largely to do with production (the creator and the process), and not with the artifacts produced or the audience which receives/uses them. Two prominent features of these arguments are:

- **The close manipulation of the medium.** Just as a traditional craftworker works directly into a medium (clay, paint etc.) without the intervention of a symbolic representation, so too the software worker creates directly with the “stuff” of the product. (Although software is clearly an allographic medium, existing primarily in notational form – like music – rather than an autographic medium in which each product is unique, and exists only in the original). In fact, with software, this argument takes a further step, as programs can become the raw material for new programs and, to the extreme, also tools from which new programs are fashioned. (Jones, 1988) That is all our materials can become tools and all our tools are raw materials.

- **The situated, contextual nature of UI “problems”.** It is difficult to frame UI problems in an industrial-design-manufacture setting. UI problems are often difficult to specify, and fall into the category of Rittel and Webber’s (Rittel & Webber, 1973) *wicked problems* – that is a problem whose formulation is necessarily vague and whose optimal solution cannot practically be found or measured. *Wicked problems* are “essentially unique” similar problems (and their associated solutions) can guide towards a solution, but cannot guarantee it: similar problems do not necessarily lead to similar solutions.

Craft & Patterns

The expression of craft is in a complete and complex product, from which it is impossible to isolate a single facet and say “this is the essential bit”. Looking at UI design as a craft skill in this way, the problem/solution pairs which form the basis of most UI patterns take on a different complexion. It is no longer “obvious” what the practices *are* that should be captured.

- Is it component-level widgets deployed at implementation to addresses a specific need? (see Martijn van Welie’s *Wizard* or Hong and Landay’s *Custom 3-D Action Buttons*)
- Is it principles that guide the use of the given functions? (see Jennifer Tidwell’s *Toolbox*)
- Is it codified examples of higher-level principles? (see The Brighton Usability Group)

Are patterns recipes? Cooking is clearly a craft skill, with a longer history than UI design, and is practised within a context of many experts transferring their expert knowledge in the form of recipes. But recipes don’t address the craft process either and a recipe is not a pattern, and a cookbook is not a pattern language.

- Recipes are authoritarian (they are a call to authority: there is a right way to make lobster thermidor, whether you like dairy products combined with shellfish or not) At this level they can be seen as component-level widgets deployed at implementation to addresses a specific need
- Cookbooks expect you to expect knowledge in prescribed ways which don’t match the order of acquisition of expertise :

“One of the differences about the universe of cooking as portrayed in beginner’s cookbooks and as we acquire it in real life is that in the former knowledge progresses in an orderly fashion, while in real life it arrives in unique chunks of experience ... and those in no particular order. In this regard, it is more like doing a jigsaw puzzle: putting your hand on just the right piece can link several other unconnected-seeming pieces together in a coherent pattern” (Thorne & Thorne, 1998)

Perhaps patterns should seek to illuminate this problem space of the craftworker, rather than the solution space of the designer.

Pointers towards appropriate Capture of Practice

- You can’t write a (craft-skills) pattern without addressing the wholeness of the artefact. You can’t address a single aspect of craft production without reference to the whole product. You can’t isolate a part of a Shaker chair and say “This shaped leg makes it special” at best you can say “The relationship of this shaped leg to this shaped chair-back makes it distinctive” (in these circumstances)
- The best patterns have this quality. They offer the craft practitioner knowledge that furthers their understanding of, and skill in, the process (practice) of their craft. They are distinguished by avoiding implementation. They speak to the “real world” of UI design where problems do not come graded, and the practitioner does not acquire the skills and experience to deal with them “in an orderly fashion”. It is the “real world” of wicked problems where solutions can only be suggested by other solutions; not determined by them.
- Craft-skills patterns, reflecting completeness in design by their very nature of their composition, may work to change attitudes in the craft workers who use them: Good patterns may more closely represent the second of the two stonemasons in the old anecdote: Two stonemasons, standing side by side are both cutting perfect 2-foot cubes of marble. The first one says “I am cutting a perfect 2-foot cube of marble” the second “I am building a cathedral”

References

- Fincher, S. (1999a). Analysis of Design: an exploration of patterns and pattern languages for pedagogy. *Journal of Computers in Mathematics and Science Teaching: Special Issue CS-ED Research*, 18(3), 331-348.
- Fincher, S. (1999b). *What is a Pattern Language?* Paper presented at the Interact'99, Edinburgh.
- Fincher, S. (2000, 7th September 2000). *The Pattern Gallery*. Available: <http://www.cs.ukc.ac.uk/people/staff/saf/patterns/gallery.html>.
- Fincher, S., & Windsor, P. (2000). *Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design*. Paper presented at the CHI'00, The Hague.
- Jones, J. C. (1988). Sofotechnica. In J. Thackara (Ed.), *Design After Modernism: Beyond the Object* (pp. 216-226). New York: Thames and Hudson.
- McCullough, M. (1998). *Abstracting Craft*. Massachusetts: MIT Press.
- Rittel, H., & Webber, M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*, 4(2), 155-169.
- Thorne, J., & Thorne, M. L. (1998). *Outlaw Cook*. Devon: Prospect Books.
- Wroblewski, D. (1991). The Construction of Human-Computer Interfaces Considered as Craft. In J. Karat (Ed.), *Taking Software Design Seriously: Practical Techniques for Human-Computer Interaction Design* (pp. 1-19). New York: Academic Press.