Some Roles of Patterns and Pattern Languages in the Capture and Transfer of Design Knowledge

Abstract

This paper examines some aspects of Patterns and Pattern languages that make them especially well suited for the capture and transfer of design knowledge. Three particular roles for the use of these representations are then identified and illustrated: 1) the capture and sharing of local expertise, within situated communities of designers; 2) the sharing of design knowledge between designers in different disciplines, and 3) the use of patterns and pattern languages as an educative tool.

1. The Relationship of Patterns & Pattern Languages to Expertise

Patterns and Pattern Languages are intimately concerned with the representation of design expertise. Some of the ways in which they do this are readily apparent; some are more subtle, embedded in the process of their construction and use. In this paper, I shall first describe patterns, using the most comprehensive and well-known example of the genre. Then I shall explicate some of the special characteristics by which they represent the nature of expertise. This is followed by examples of how patterns are actually used, dividing these into three distinct roles in which experts employ them. Finally, I shall discuss some aspects and properties that relate not to their representation of expertise but to the transfer of the knowledge they contain, to the nurture of experts.

2. Alexander's Patterns

Patterns (as such) were defined and named by Christopher Alexander in his two works *A Timeless Way of Building* (Alexander, 1979) and *A Pattern Language* (Alexander, Ishikawa, & Silverstein, 1977) within the domain of architecture specifically and the built environment generally. They espouse an approach to design—codified in the patterns—that focuses on the interactions between the physical form of buildings and the way in which that form inhibits or facilitates various sorts of personal and social behaviour (Bayle et al., 1998). Important aspects of Alexander's patterns are:

- they were devised with the express intention of providing a common vocabulary between users and architects, as well as among architects themselves
- patterns are not created or invented; they are identified via an invariant principle (of good design) as manifest across different places and cultures (several examples are given in each pattern).
- they are structured around the problems that designers face, and those problems are addressed by the provision of a "solution statement". As such patterns are embedded in practice: addressing recurrent problems with invariant solutions: "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" (Alexander et al., 1977) p. x

Each "pattern" follows a prescribed form that is based on evidence for, and examples of, the use of the pattern, together with instructions for how to achieve its effect. Each pattern is named, and illustrated with both a photograph and a diagram. The 253 patterns that Alexander identifies are collected together into a pattern "language", which allows them to be

used in combination with other patterns (often at different scales) so that whole environments can be constructed along these principles.

A PHOTOGRAPH	showing an archetypal example of the pattern in use
AN INTRODUCTORY PARAGRAPH	which sets the pattern in the context of other, larger scale patterns
THE HEADLINE	An encapsulation of the problem (one or two sentences)
THE BODY	Of the problem (this can be many paragraphs long)
THE SOLUTION	the heart of the pattern, always stated in the form of an instruction
A DIAGRAM	Shows the solution in the form of a diagram
A CLOSING PARAGRAPH	Shows how this pattern fits with other, smaller patterns

Alexandrian Pattern Format. The sections and descriptions are from (Alexander et al., 1977), the comments in brackets mine

Each part of this form has an impact on the transfer (and transferability) of the design knowledge it contains:

Name: The name of a pattern is considered to be very important, indeed, it had been given greater prominence in pattern explorations subsequent to Alexander's. This is, I believe, because as patterns are used, then the knowledge that they embody can be "short-handed" by simply referring to the name: a design synecdochism.
Photograph: A common part of pattern presentation, and a very strong one, is the inclusion of a concrete example of implementation of the pattern. In Alexander, a photograph conveys this example of implementation. The importance of this part of the pattern template is not immediately obvious, but I believe that the purpose of this component is to sensitise the reader to the application of the pattern. These "sensitising examples" invoke a reaction in the viewer. The intention is that the reaction is favourable—"Wow, that's good. I'd like to live there"—and from that point the reader is sensitised so that the information that the rest of the pattern contains becomes more accessible, more useful in a specific implementation.
Situation within Pattern "Language" (introductory and closing paragraphs) These are "way-finding" devices that take the reader to other relevant, related aspects of the design space. In Alexander the principle on which these are structured is scale (from "city" to "house", for example), so it is reasonably easy to conceive of patterns combining together to form larger structures. (I believe there are considerable problems for structuring principles in other pattern languages, and have explored this issue at length elsewhere (Fincher, 2002))
Problem ("headline") and Solution: This symmetrical construction is a central part of a pattern's form, and is both a unique and a strong way of presenting the knowledge they capture. Because pattern languages are about the design of environments (be they architectural, interactional, software or pedagogic) anchoring the knowledge to the practice is key. At best the "headline" presents a problem that is frequently occurring (and therefore easily recognisable). Problems are often quite general. The solution tells you how to solve the problem. Solutions are always presented in the form of an instruction, so the reader (as designer) is never in doubt about what to do. In implementing the solution (or even thinking about it) a designer will not only add to their repertoire of techniques, but will also learn to recognise the "hot-spot" problem areas in their design space.
Body: this is the place in the pattern that provides the rationale, examples and references that form the body of knowledge from which the pattern is harvested.

3. Some Roles of Pattern Languages

Since Alexander's specific construction, other groups have tried to replicate the approach in other domains (notably software, see: (Gamma, Helm, Johnson, & Vlissides, 1994) (Gabriel, 1996) latterly pedagogy (*The Pedagogical Patterns Project*, 2001) and more recently, for "shaping the network society" (*DIAC*, 2002)). Less common are accounts of how patterns and pattern languages have actually been used. In terms of design, and design expertise, I believe there are three groupings which demonstrate different roles pattern languages have (or can) play in design and design education.

3.1. Capturing local practices

Most of the published pattern languages are grand endeavours. They try to encapsulate knowledge of how to build a "city" or an "interface". Most designers are engaged in smaller-scale activities, which are constrained within a reasonably small domain of application. In these more local settings, there is a use of patterns to capture practice and values –"the way we do things here" so that they can be transferred to newcomers to the organisation, or used as a yardstick when discussing other alternatives, other peoples' products and practices. I shall discuss three examples of this sort of activity.

a) The Océ Experiment

From 1997 to 1999 Joerka Deen from the Industrial Design department of the photocopier firm Océ set up a "Pattern Committee" to document design practice (Deen, 2000a, 2000b). His work was guided by the specifics of his situation: an over-arching aim of Océ was to focus on "harmonizing the human experience" across all aspects of their product, providing a seamless interaction experience. However, as in most Industrial Design departments, they did not have complete control over the conceptual design decisions that went into the product—but they could influence all such decisions. He says:

> So we are faced with the challenge to influence multiple autonomous project teams from within the project team, in order to harmonize the human experience which is not a goal of an individual project team. So could one begin to get some grip under these circumstances?

He addressed this dichotomy with patterns:

> Alexander's work inspired me to set up a Pattern Committee with Interaction, Graphical, and later on, Product Designers. I was a strong believer in Alexander's way and we naïvely started a daunting task. We spent every other Thursday, a full afternoon with about seven people, trying to create patterns

They eventually created a set of Océ-specific patterns: a photocopier-interface pattern language. Their patterns all have names and associated images, but the remaining pattern components are fluid and vary somewhat between examples.
Deen says that this pattern-creation effort had an immediate effect on knowledge, in that it created a shared vocabulary within the project teams which allowed communication of tacit knowledge and the documentation of explicit knowledge. Over time, he claims, it also showed returns in the efficiency and quality of the product designs.

b) The Swanwick Experience

In the year 2000, work on a new en-route air traffic control centre had been underway for 9 years. In 2000 a second centre was commissioned in Scotland. The two centres were to use a common system and hence a common user interface, but the unique features of the Scottish airspace meant that additional functionality had to be developed and the user interface

evolved. A separate team were to undertake the Scottish work. Peter Windsor (one of the original Swanwick architects) describes their situation in this way (Windsor, 2000):

> Although the user interface is specified in considerable detail, the principled rationale for the design choices has only been documented in part. Thus, this knowledge has remained the collective property of the user interface design team. Long term, however, this is, of course, undesirable:
>
> - there is a substantial learning curve for new members of the team to understand the rationale for the user interface before they can be productive;
> - the expertise is liable to be lost over time, so future evolution of the system may inadvertently reduce its usability.

He continues:

> Our remit … was to produce a project style manual: a project specific set of user interface design principles and guidelines. However, we got stuck on the two well-known limitations of this approach:
>
> 1. broad principles, such Nielson's usability heuristics, are open to a range of interpretations and are difficult to make domain and project specific;
> 2. detailed guidelines, which can be domain and project specific, are applicable in some contexts but not in others.

The team responded to these challenges (and reconciled them) by creating a small pattern language. Because this was for their own conditions with very constrained needs of transfer between small and knowledgeable communities, they used a minimal pattern-form (name, problem statement, solution statement with embedded links and supporting examples) and structured them around the characteristics of their users' tasks: context, trigger event, frequency, duration, interrupt strategy and tolerance to errors.
When they had completed this Windsor noted that whilst they certainly captured their experience and expertise as patterns they also had "a whole bunch of stuff" left over which was "classic design-guideline material". The pattern language component was described as useful, and particularly something that represented "the way I and my colleagues do design"

c) Berkeley CA: JSW Architects

25 years after the publication of *A Pattern Language* two of the original authors (Max Jacobson and Murray Silverstein), plus a third (Barbara Winslow, their partner in architectural practice), published *Patterns of Home: The Ten Essentials of Enduring Design* (Jacobson, Silverstein, & Winslow, 2002). Clearly more limited in focus than the 253 patterns encompassed by the earlier work (they are concerned only with the design of domestic, residential accommodation) they offer another reason for their collection:

> Best of all, because the authors have spent most of their professional lives using and evolving the patterns, they bring a breadth of knowledge that gives the reader access to a quarter century of refinement of the original notions.

What this results in is not a book of many patterns, each illustrated with a "killer example", but a more closely textured book, where a small number of design patterns (10) are played and re-played. Their pattern form is almost non-existent; just a name and a description. It is a book not of what the patterns are, but of how they can be instantiated in practice.
And this is one of the noticeable things about the collection, the (not unpleasant) sameness of the houses. Most are situated on the west coast of the USA, are large in scale and apparently large in cost. Nothing is wrong with any of this, they are beautiful houses and probably easy, comfortable and comforting to live in. What they represent is the expertise and values of one

set of designers situated in a constrained local context, the context here being not an application domain, but one of geography and culture.

These three examples all show small teams of designers, radically co-located and working within very specific domains, using patterns to capture and communicate their expertise. Sometimes (as in Océ and Swanwick) this is for the purpose of documentation, for explicating the important parts of practice to pass on to other team members. For JSW the audience is wider, but is still composed of people who want to act as designers within the same domain.

It is notable that all of these examples use a very minimal form in which to express their material. Partly this is because none of them have sufficient patterns to be concerned with structuring them into a "language", so don't need the wayfinding aids to other patterns. However, in particular, none of them have the "body" section of the traditional pattern form – the background that gives the reader the reasons why this practice is good and useful and where to find out more about it. Perhaps this is because their pattern intent is carried as much by the very tight application areas they are working within as by the format of the documentation; perhaps it is because they are using the form to communicate with other experts – who might be expected to be able to supply the reason they are using a pattern for themselves. In either case, the pattern authors are relying on having a shared understanding of the domain with their anticipated audience. In this way all of them are fundamentally concerned with creating and sustaining a community of practice.
PATTERNS AND PATTERN LANGUAGES WORK in this role by the facility with which they can build on shared disciplinary structures to express the local values which inform the design practices of the teams.

3.2. Brokering design knowledge between disciplines

The most mature expression of the potential for patterns and pattern languages as knowledge structures that bridge designers' understanding is presented by Tom Erickson in his paper *Lingua Francas for Design: Sacred Places and Pattern Languages* (Erickson, 2000) . He states the need for a *lingua franca* quite plainly:

> … the concept of a *lingua franca*, a common language which is accessible to all the participants in a design process. By common language I am referring not just to vocabulary (otherwise English or another natural language would typically serve quite well), but to the conceptual frameworks which disciplines and professions bring to bear during the design process. The idea to be explored here is that part of the process of interactive systems design should be the development of a *lingua franca* for a particular design project, a common language which permits all stakeholders to participate more fully in the design process.

He illustrates this need with an extended example concerning the work of urban designer Randolph Hester in the town of Manteo in North Carolina. Hester was brought in to work on a plan for achieving economic renewal without sacrificing the town's character. One of the things that he and his team did was to map the "sacred structures"; the places that the residents valued, the places which made Manteo the sort of town they wanted to live in. Interestingly, "these places [were] almost universally unappealing to the trained professional eyes of an architect, historian, real estate developer, or upper-middle-class tourist." and "only two were protected by historic preservation legislation … that is, the existing planning and legal mechanisms that were intended to help preserve the character of places missed most of what the residents of Manteo actually valued".

Not only was the resultant list of "sacred structures" a driving force for the re-development at the time, it was still being used and referred to seven years later. As Erickson concludes, "this is an amazing and inspiring result, perhaps the highest goal to which a designer can aspire. Hester's work in Manteo resulted not only in a plan for achieving economic renewal without sacrificing the town's character (the explicit goal he was employed to achieve), but it also resulted in a *shared, self-sustaining system of beliefs and values* that enabled the plan to be realized over a much longer period of time". (original emphasis)
In the on-line version of the paper, Erickson appends a short section where he engages with questions raised by reviewers. In this, he identifies some specific features of patterns and pattern languages that, he believes, make them work as *lingua franca*:

> *Why do need a special lingua franca? That is, why is it not sufficient to use language and discussion?*
>
> … to me the answer has to do with the nature of disciplines and expertise. When designers get involved in a project, they bring with them a conceptual framework that has, often implicit in it, perspectives, values, methods and so forth. When the design team is itself interdisciplinary, conflicting conceptual frameworks can lead to discord and confusion. But even if the design team has a single conceptual framework that they apply, the users are at a great disadvantage: they are confronted with designers speaking a language that—though full of words everyone understands—refers to concepts, methods, values and assumptions that arise from a discipline or profession rather than from the users' daily lives. The example of Manteo illustrates the power of rooting design in concrete exemplars of the design domain, rather than abstract disciplinary based concepts.

PATTERNS AND PATTERN LANGUAGES WORK in this role because of an aspect of the form (particularly as expressed in the "complete" Alexandrian format): the inclusion of both the principle of good design that the pattern is attempting to convey *and* a specific example of an implementation of the pattern in a real artefact.

Why does this work?

In general you can make something simple in one of two ways. You can apply general principles across situations – mostly this is the aim of formal education, to present principles illustrated with situations in which they apply, trusting that students can then make future conceptual bridges for themselves (that is, correctly apply the principle to unknown situations). Or you can make things tightly specific – mostly this is what students try to do with assignments, they want to know *exactly* how to do *exactly* this thing, and no more. (It's possible that this tension is a more general description of the acquisition and deployment of expertise). One of the very powerful things about patterns is that they address both views at the same time. They provide the general principles *and* a set of precise instructions as to how to implement and achieve it.

3.3. For education

There is a third role which pattern languages can take with regard to expertise, and that is one of education.
In 1978 Edwina Rissland Michener (Michener, 1978) wrote *The Structure of Mathematical Knowledge* (SMK): its similarity to Alexander's *Pattern Language* (published 1977) is most striking.

First there is a similarity of overall form: as with patterns, each individual part of the framework (here called an "item") has a common format, and these "items" are brought together into a whole via a structuring principle (in this case a series of structuring principles).
Secondly, the presentational format also shows considerable similarity: particularly noteworthy is the inclusion of "star ratings", pictures and references to "predecessors and successors" within the structure

1. HEADER information, such as ID, NAME, epistemological CLASS the Michener Rating (from one to four *'s) which describes the importance of the item relative to the theory as a whole, and other high-level descriptors.
2. STATEMENT information which explicitly declares the mathematical SETTING and includes the declarative formulation of the item:
    a. in the case of a result, its if-then STATEMENT;
    b. in the case of a concept, its mathematical DEFINITION;
    c. and for an example, a CAPTION stating what the example illustrates.
3. DEMONSTRATION information which includes the procedural aspect of an item:
    a. a PROOF in the case of a RESULT;
    b. CONSTRUCTION for an example
    c. and a PROCEDURAL formulation for a concept.
4. a PICTURE which is static or dynamic (i.e. sequence of) pictures
5. IN-SPACE POINTERS to the item's predecessors and successors
6. DUAL-SPACE POINTERS to its two associated subsets of dual items
7. PEDAGOGICAL data indicating which teachers use this item when
8. REMARKS on the item, such as when and how to use it
9. EXTRAS which fine-tune the representation for an example, concept or result
10. Additional data such a bibliographic references and useful applications

Michener "item" format

Thirdly, there is a similarity of intent. There is little purpose in capturing expertise without the idea of doing something with it. Alexander's intent is to empower, to make experts of all participants (in the design of the built environment that they inhabit). Michener's intent is to make experts of learners of mathematics

> These ideas can also be applied to the design and support of interactive environments for experienced mathematicians and neophyte mathematical students. GS (the GROKKER system) and GLA (GROKKER LEARNING ADVISOR) … are examples of the sorts of system we envision. GS is designed to help professional mathematicians retrieve and explore mathematical domains. GLA is an advisor program to be used in conjunction with GS to allow neophytes to work in the manner of experienced mathematicians and help them to learn how to understand, in short, to learn as some expert students do

PATTERNS AND PATTERN LANGUAGES WORK in this role because of the relationship between the pieces that are patterns and the structure that relates them. One characteristic of expertise is that bits of knowledge are not isolated, but related to one another, anchored in larger conceptual structures. By virtue of the explicit situation of a specific piece of practice in the context of a larger whole, a newcomer learns the knowledge in the pattern, certainly, but also—and at the same time—what else it relates to in the design space.

4. Summary thoughts

I believe patterns are particularly useful, and effective, for the transfer of expert knowledge (and hence the nurture of expertise) because there are a dense and layered representation.

- As we have seen the problem/solution format combines *abstract and concrete* recommendations
- The situation of the patterns within a *structuring principle* allows each piece of practice to be understood within terms of a larger whole, but that whole is also illuminated with the addition of each pattern understood.
- The inclusion of a pattern within a pattern language is neither capricious nor arbitrary. It is there because the authors consider it to be representative of good design. By its existence it tells you *what experts think is important* in the design space, it tells you the values that inform their practice.

This density of representation, the combination of design values, structure of the design space and specific instructions for implementation mean that the acquisition of expertise is eased. Transfer of knowledge is hard in any situation. The presentation and, especially, the implied value system means that a pattern language is not just an expression of expertise, but a *purposeful* one. It means the transfer is not arbitrary, you know *why* you are learning as well as *what* you are learning, it is a multi-dimensional, layered approach which reflect human needs and human practice: it is humane.

References

Alexander, C. (1979). *The Timeless Way of Building*. New York: Oxford University Press.

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Constructions*. New York: Oxford University Press.

Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, B., et al. (1998). Putting it all Together: Towards a Pattern Language for Interaction Design. *SIGCHI Bulletin, 30*(1), 17-24.

Deen, J. (2000a). *CHI 2000 Workshop Position Paper*, from http://www.it.bton.ac.uk/staff/rng/CHI2K_PLworkshop/Participants.html

Deen, J. (2000b). *A Patterns Language for Interaction Design*, from http://sigchi.nl/activiteiten/verslag-20001212.html

DIAC. (2002). from http://www.cpsr.org/conferences/diac02/

Erickson, T. (2000). *Lingua Francas for Design: Sacred Places and Pattern Languages.* Paper presented at the DIS 2000.

Fincher, S. (2002). *Patterns for HCI and Cognitive Dimensions: two halves of the same story?*. Paper presented at the Fourteenth Annual Workshop of the Psychology of Programming Interest Group, Brunel.

Gabriel, R. (1996). *Patterns of Software: Tales from the Software Community*. New York: Oxford University Press.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Massachusets, US: Addison-Wesley.

Jacobson, M., Silverstein, M., & Winslow, B. (2002). *Patterns of Home: The Ten Essentials of Enduring Design*. Newtown, CT: Taunton Press.

Michener, E. R. (1978). *The Structure of Mathematical Knowlege.* Unpublished PhD, MIT.

The Pedagogical Patterns Project. (2001). Retrieved November 2001, from http://www.pedagogicalpatterns.org/

Windsor, P. (2000). *A Project Pattern Language for Interaction Design.* Paper presented at the BCS HCI Group/IFIP Working Group 13.2: Workshop on HCI Patterns.