

Putting It All Together

Towards a Pattern Language for Interaction Design: A CHI 97 Workshop

Elisabeth Bayle, Rachel Bellamy, George Casaday, Thomas Erickson, Sally Fincher, Beki Grinter, Ben Gross, Diane Lehder, Hans Marmolin, Brian Moore, Colin Potts, Grant Skousen, John Thomas

Pattern languages are representations that have been used in architecture and urban design for about twenty years. They focus on the interaction between physical form and social behavior, and express design solutions in an understandable and generalizable form. But pattern languages are not simply set of patterns intended to be universally applied; instead, they are actually meta-languages which, when used in a particular situations, generate situated design languages. This report describes a CHI 97 workshop which explored the utility of pattern languages for interaction design. We discuss the workshop's rationale, the structure and process of the workshop, and some of the workshop's results. In particular, we describe some patterns developed as part of the workshop, and our consequent reflections on the use of patterns and pattern languages as lingua franca for interaction design. This report concludes with a bibliography on pattern languages and related matters that spans architecture, software design, and organizational design.

Introduction

The Challenge: Complexity and Diversity

Interaction design is becoming an increasingly complex and diverse activity. It is becoming more complex in that communications and computational technologies, as they become cheaper and smaller, are being integrated into more devices and embedded in more environments. This, in turn, makes interaction design relevant for an increasing number of new application domains. And even as the space of design possibilities increases, workplace studies are making us aware of the complexity of the socio-technical systems within which we are working to integrate new technologies. How are we to manage this complexity?

Interaction design is becoming more diverse in that a wider range of people are becoming involved in it. Within CHI, it is well accepted that anthropologists, psychologists, and visual designers, as well as engineers and computer scientists, have roles to play in systems design. As computing systems shrink in size, industrial and product designers need to work hand in hand with systems designers. The advent of virtual spaces create roles for architects and interior designers. The commercialization of video and multimedia technologies create roles for musicians, film producers, et al. While the multidisciplinary nature of interaction design brings much richness, it is also challenging because no common perspective, set of practices, or theoretical orientation can be assumed.

Another factor driving the diversification of interaction design is customization. As systems become increasingly customizable, more and more design – in the sense of front end creation, application programming, and software configuration – is being done in-house. Sometimes this means that traditional MIS departments are playing a role; sometimes it means that external consultants are involved; sometimes it means that end users themselves participate. In many cases these participants lack the time, resources, or inclination to engage in research on the needs and practices of their users. And, in many cases, these participants lack formal training in design, and hence any common perspective or language.

A Possible Solution: Pattern Languages

So, we have a rapidly expanding game: more players and more technology projected onto workplaces which we are learning more and more about. This increasing complexity and diversity can

be source of richness, or of chaos. Thus, we need to explore ways of dealing with the increasing complexity and diversity of the interaction design field. This workshop explored one approach to putting it all together through a common language. Our model is the work of Christopher Alexander and his colleagues who over the last few decades have looked at what works and what doesn't work in architecture and urban design. The basic approach is to closely examine particular cases, attempt to identify recurring patterns and integrate them into a language of relatively concrete patterns. Their work is codified in the book *A Pattern Language* [1]: each pattern is described; examples are given; empirical data supporting the pattern are referenced; and the relationships to other patterns are defined. The way of using the patterns, that is, the process of design, is described in a companion volume, *The Timeless Way of Building* [2].

Let's take a brief look at Alexander's Pattern Language. The language consists of a network of over 250 patterns. The patterns cover a range of scales, from a pattern for the distribution of towns and cities to patterns for walls and room sizes. The patterns are loosely connected across scales: any given pattern typically points to smaller scale patterns which can support it, and larger scale patterns in which it may participate. For example, a pattern called 'Identifiable Neighborhood' (aimed, obviously enough, at creating neighborhoods with their own particular sense of place) will involve a number of lower level patterns. Possibilities include 'Street Cafe', 'Individually Owned Shops', 'Corner Grocery', 'Beer Hall', and so on. At the same time, 'Identifiable Neighborhood' partici-

pates in larger scale patterns that specify characteristics of communities.

Now let's look at a few examples of individual patterns, though these short summaries do not give the full flavor of the careful analysis and data that are contained in the complete patterns. 'Eccentric Neighborhood Centers' points out that neighborhood centers should be off-center; that is, closer to downtown because people will tend to go toward the city center rather than away from it. 'Beer Gardens' suggests that community pubs should have activity around the edges and large tables in the middle. This encourages people to cross through the center and sit at the tables. By contrast, many modern bars have very small tables, and it becomes uncomfortable for a stranger to casually approach another because the space is too intimate. Thus, the opportunity for the pub to serve as a cohesive force is diminished. 'Gradient of Privacy' says that there should be a gradual gradient from public to private space in a house; e.g., from porches that look on the street life to entry ways to public rooms to family rooms to the bedroom.

These patterns focus on the interactions between the physical form of the built environment and the way in which its form inhibits or facilitates various sorts of personal and social behavior within it. This emphasis on the overlap between the physical and behavioral worlds brings to mind the Gibsonian concept of affordances, although what is going on here is much subtler: whereas saying that certain properties of the environment afford certain actions simply means that it makes them possible (most tables and chairs afford being stood on by a person), the Alexandrian emphasis is on characteristics of the environment which facilitate or inhibit (the presence of a table and chairs can facilitate people working together or sharing food).

While these patterns describe basic ways that space should be organized in order to have a positive impact on human feelings and behavior, we believe there is a possibility to create something analogous for processes of human interaction. As representations, pattern languages have very interesting properties:

- they are based on concrete prototypes drawn from the domain in which design is being done
- they work at multiple levels – community, group, individual – and they endeavor to tie the levels together
- they attempt to bridge the gap between the physical and social worlds
- they seem to be amenable to gradual, piecemeal development

We believe that these properties might enable pattern languages to serve as a *lingua franca* for the diverse community of interaction designers.

(It should be noted that there is considerable interest in pattern languages as vehicles for object-oriented software design reflected both books (e.g., [4, 5, 6]), and in conferences and mailing list activity (see the Patterns home page at <http://st-www.cs.uiuc.edu/users/patterns/>). However, as yet, there has been little attention given to pattern languages for interaction design.)

The Structure of the Workshop

Goals of the Workshop

The workshop had a number of goals. The first was simply to get an idea of where the state of the HCI community *vis à vis* pattern languages: Were pattern languages simply an attractive idea, as yet untried in the domain of interaction design? Or were pattern languages being created and used? If so, how were they being used, and to what ends were they being applied? Second, we wanted to share whatever knowledge and experiences workshop participants had, so as to deepen our own understanding of patterns. Finally, we wanted to investigate how we might further the exploration of pattern languages within the HCI community.

Preparation

The preparation for the workshop followed the usual route. We put out a call for participants in the CHI Conference announcements, and also on a few mailing lists that reached other relevant communities. We asked applicants to submit position papers, and ended up with 13 participants, one more than the even dozen we had set as our goal. Prior to the workshop we used email to develop an agenda, and to distribute the position papers to the group.

Workshop Activities

The workshop consisted of a mixture of activities. The goal for the first day was to get the variety of ideas and perspectives out on the table, and to develop a shared understanding of one another's perspectives. This was accomplished through a mix of long and short presentations, interspersed with discussions. We devoted one of these sessions to the model of dialog popularized by David Bohm [3]: essentially this is a set of practices which support a reflective and collaborative exploration of individuals' mental models.

The goal of the second day was to work together to further our understanding of pattern languages. After an initial gathering in which we identified some ways of proceeding, we broke into small groups. One group categorized the themes we explored the day before. The other groups went out pattern hunting, and spent an hour and a half prowling the CHI Conference area looking for patterns. Afterwards, the groups discussed their findings, and then turned to the business of generating an interactive poster for the CHI Workshops poster session. Besides describing the workshop, we decided that we would use the interactive poster as an opportunity to give other CHI attendees an opportunity to observe, write, and share their patterns.

Lessons from the Workshop

It's difficult to encapsulate two days of presentations, discussions and activities in a few pages. In particular, this risks exaggerating the degree of certainty and consensus that was actually achieved. To partially counter this, quotes from individual participants are interspersed below. So readers should be warned that there was a range of viewpoints, and few, if any, of the claims, conjectures, or cautions we offer were held with complete unanimity.

With these caveats in mind, let's turn to the results of the workshop.

Perspectives and Interests of Workshop Participants

Not surprisingly, the participants in the workshop represented a diverse range of

views and interests. Here we try to capture some of this diversity:

Two Meanings of 'Pattern'

We used 'pattern' in two senses:

- **Activity Patterns:** Patterns that describe things as are they are, without making a judgment as to whether the pattern is worthy of emulation or preservation.
- **Design Patterns:** A pattern that describes a connection between a repeatedly encountered problem and a solution that has been proven in the field, across time and circumstance.

Naturally, our different uses of 'pattern' tended to reflect the uses we had in mind for patterns.

Many Ways of Using Patterns

Participants were interested in using patterns in many different ways (some complementary, some contradictory):

- **Capture and Description:** Patterns can be used to describe the key characteristics of a place, situation or event, in a context-sensitive way.
- **Generalization:** Patterns can be used to support generalizing across places and situations, while retaining an element of concreteness.
- **Prescriptive:** Patterns (i.e. design patterns) can be used to prescribe solutions to commonly encountered problems in a particular design domain. Thus, patterns might be used as a way of presenting HCI guidelines, or guidelines for organizational design.
- **Rhetorical:** The concreteness of patterns, and the fact that they are drawn from the situation for which design is being done, makes them very appropriate as a *lingua franca*, a common way of talking about design issues that is accessible to designers (of whatever disciplinary backgrounds) and the users and inhabitants of the situation.
- **Predictive:** Patterns can provide a 'what-if' mechanism for reflecting on the possible impact of changes to a place or situation (by looking at the patterns a change would directly impact, and by tracing ramifications of such a change through the network of patterns).

Application Domains

- And workshop members were interested in applying patterns to a wide variety of domains:
- Embodying HCI Guidelines as patterns
- Using patterns for process and organization design
- Exploring how Alexandrian patterns (as documented in *A Pattern Language*) might be applied to new situations (e.g., cyberspace)
- Establishing pattern-making as an organizational practice
- Using patterns to clearly and succinctly describe particular workplaces, in order to understand possible impacts of new technologies.
- Using patterns as a tool in participatory design.

Some CHI Conference Patterns

In this section, we describe some of the patterns we observed at the CHI Conference. Most of these arose from our workshop session, although a few patterns that came from the poster are included. Since the workshop was held before the conference, the patterns tend to be associated with the CHI Registration area, which is where most activity was happening.

All of the patterns here are necessarily 'activity patterns' – that is, these patterns are simply what we observed and reported. They may or may not be 'design patterns', that is, proven solutions that should be emulated in the future, and in other contexts. Given that the organization of the CHI Conference reflects years of trial and error and optimization of solutions, it is likely that some of the patterns *are* design patterns – we just don't know enough, based on our brief investigation of a single context, to say which are and which aren't.

Administrative Nexus in the Eddy

Problem Context: The conference management office is the heart of the conference. It is the location from which the conference is run, where the key personnel can most often be found, and the center from which crises are likely to be resolved. It is important to select a good location for the conference management office.

Forces: The conference office should be located near the loci of most frequent (or severe?) problems. It should also be easy to mobilize student volunteers, and other resources, from the management office in the event of crises. At the same time, the management center should not be so prominent that it attracts people with casual requests, drop-in visitors, or other non-crucial interruptions.

Solution: Locate the conference office near – but not too near – the conference registration area.

Comments: This pattern was based on observing the location of the CHI Conference Management office, and the activities which occurred there. The office was located near the reception area, but far enough away to not be noticed by those who were not looking for it. There were lots of student volunteers nearby in the registration area, and we thought it likely that the registration area was a source (or at least attractor) of problems.

Divide and Conquer Queue

Problem Context: Lots of people arrive to register at once, often not long before an event.

Forces: Many people are frustrated by standing in long lines, and are particularly impatient if an event is about to begin. At the same time, people don't like to see others who arrive after them, being served before them.

Solution: Given that there are plenty of student volunteers, and ample space, divide the registration queue into smaller units according to well-defined organizational principles (e.g. alphabetical order). If a particular line is markedly slower than the other, the dividing algorithm can be tuned by expanding/contracting the alphabetic slots.

Comments. A particularly nice aspect of this instance of the pattern is that the registration packets are organized by the same (alphabetic) principle as the lines, and so the packets can be spatially divided (and re-divided) to match the composition of the queues.

Clarification Graffiti

Problem Context: Designers try to anticipate interaction patterns and add

cues ahead of time to facilitate interaction. However, since the perspective of the users in use is different from that of the designer, the cues are never sufficient.

Forces: Signage (and other cues) should be consistent to signal their relevance to the user, and to be aesthetically pleasing. Of course, this requires that signs are built ahead of time, outside of the context of use, and thus guarantee insufficiency.

Solution: Allow users to add annotations to pre-designed cues.

Comments: This pattern was based on the observation of a preprinted sign, near the elevator, that said 'Message Board on Ballroom Level.' Below it, someone had taped a handwritten note that said '(Go up one floor)'. Evidently, the sign designer was more familiar with the hotel and its named spaces than many of the conference attendees. (This pattern is further discussed in the section 'Values and Pattern-Making').

Observations and Conjectures about Patterns

A Continuum of Patterns

There is a continuum of patterns relevant to HCI from very high to very low level. Some of us had initially imagined workplace pattern languages, organizational pattern languages, and user interface pattern languages as very different things. However, by the end of the workshop it seemed difficult (and inadvisable) to distinguish among them. At the same time, we did feel a need for structuring the continuum, and did feel that there were differences between patterns at various levels. One conjecture was that the higher level patterns are more robust over time, whereas lower level patterns will change more quickly, in synchrony with technological evolution.

Comments:

I now see a continuum from patterns for learning organizations down to patterns for laying out dialogs. I feel like I better understand the relationships and the dependencies between them.

[Is there] a need for a broader classification schema derived from the multitudes

of patterns that exist? During the two days we talked about organizational patterns, software patterns, interface patterns, [and many other types of patterns]... I expect that each one of these groups can be broken down – how can we merge or span different pattern languages? (I think that there might be an analogy here between the variety of architecture description languages that exist within the software architecture field, and the attempts now to find languages that span languages...)

The Centrality of Values

Values tend to be expressed explicitly in patterns. An interesting commonality we discovered is that virtually everyone agreed that values were a central part of design. Someone said, 'design is not about truth, but about values.' We speculated about the degree to which this assumption was shared by interaction and software designers in general. Most of us believed that we were in the minority, and that pattern languages – by virtue of their explicit representation of values – attract designers who think in terms of values. On the other hand, it may be that a concern with values is more widespread than we thought, but that this concern is masked by the academic orientation of CHI, and the domination of the rhetoric of usability.

Comments:

The extent to which patterns (at least design patterns) are driven by goals which are in turn determined by valued results was something I had not quite thought of that way before. It probably explains Alexander's outsider status in architecture, since his values do not seem to match the needs of large scale architecture.

Patterns and Time

Many interaction patterns have a strong temporal component, as well as a geometric component. But we don't understand, very well, how to go about capturing this.

Comment:

As I reflect on the interaction patterns we wrote and some of the discussions I realize that time is a critical factor in interaction (how long does it take for some of these patterns to unfold) and yet some-

thing that's very hard to express in patterns.

Pattern Making and the Quality of Patterns

The Experience of Pattern-Making

We found that pattern-making was an engaging and enjoyable experience. The patterns often seemed beautiful, evocative, and gave us insights into the a situation that we would have otherwise lacked (e.g. making us aware of connections between seemingly unconnected aspects of a situation). We felt that reading and discussing patterns, and especially pattern-making was, for us as designers, a valuable way of immersing ourselves in a situation. (At the same time, we had reservations about the validity and quality of our patterns.)

Comments:

The interaction patterns that emerged on the poster board had a certain beauty to them. The experience of looking at them evoked a 'that's really true' feeling of recognition. This points, I think, to the usefulness of this kind of work.

I think that just the way one is able to easily experience the feelings of the situations [the patterns] depict is valuable. If we could move IT development closer to this – that is, getting users to really work at imagining how they would feel with the system being designed – we would be a long way ahead.

Patterns are especially useful and desirable because they tend to lead to much more holistic and ecological thought processes and outcomes. A human-centered idea of the big picture is sorely needed in information systems design.

The Difficulty of Pattern-Making

It is easy and fun to begin seeing and making Patterns – but it is difficult to make good ones. A number of participants were interested in methods of making and validating patterns.

Comments:

I also came away with a better appreciation of how easy it is to write patterns and how hard it is to write ones that capture deep insights into design.

I was ambivalent about the patterns we developed. On the one hand, some of the

descriptions were very evocative, and sensitized me to aspects of the environment that I ordinarily would have been blind to (as did the process of looking for patterns). I liked the feeling of recognition that some of the patterns evoked. On the other hand, I wasn't usually satisfied with either the analysis of forces underlying patterns, or the cases where solutions were proposed.

I'm also concerned about the methods that are used to gather patterns. I can think of ways to generate patterns using qualitative methods, but how about quantitatively generated patterns...

I'm not sure (in general) how to know when one pattern is 'correct' except in the context of the whole language.

The Difficulty of Pattern-Language-Making

Another issue that was raised, though not experienced in the short time frame of the workshop, was how to go about building an entire pattern language. It took Alexander and many colleagues a decade to develop a pattern language for architecture; interaction design is a much broader domain, and the technologies that are its substrate are changing with great rapidity.

Comment:

Making a series of inter-related patterns comprising a pattern language looks like a difficult problem in "global optimization" – that is, it isn't clear to me how abstract to make patterns in order to end up with something that is tractable yet evocative.

Values and Pattern-Making

Our exercise in pattern making revealed that formulating patterns was not easy. Creating a pattern is an exercise in applying values. Consider the case of the 'Clarification Graffiti' pattern produced during the exercise. 'Clarification Graffiti' embodies the observation that users often add information to official signs. Thus, a vending machine may have a hand-written note saying that it eats quarters, or a ticket window may have a handwritten 'sold out for the 8pm show' note, or a copier may have a post-it warning about the use of transparencies.

'Clarification Graffiti' suggests that this pattern of behavior is good and useful; that is, it is a design pattern, a prescrip-

tion to which systems designers should adhere. But, there are two other formulations as well:

- It doesn't matter what designers do. Users *will* provide 'Clarification Graffiti' somehow. Therefore, it isn't really a design pattern that 'good HCI' needs to support – it's just a descriptive pattern that captures something common about human behavior.
- 'Clarification Graffiti' is not a 'feature' that should be supported; rather, it is a symptom of something gone wrong – namely, that the designers did not sufficiently test the system with real users in real contexts before instantiating the design. Thus, 'Clarification Graffiti' is really an anti-pattern: it encourages a short-term symptomatic fix when what is really needed is a long-term change in the *process* by which systems are designed and deployed.

How do we choose amongst these formulations? Use my intuitions? Yours? Is there a method which could be employed? Would such a method be applicable just to 'Clarification Graffiti', or would it require that the entire pattern language be evaluated? Is evaluation really even the issue here? It seems as though these alternate takes on 'Clarification Graffiti' are entwined with values, politics, realities of practice, and other highly situational, subjective issues. Perhaps the value of patterns here is not that it provides a way of choosing the 'right' reading, but rather that it raises these issues and can make them a subject for discussion among the stakeholders and users of the to-be-designed system.

Pattern Making as a Collective Activity

Building a Pattern Language for HCI needs to be a wide-spread collaboration. This is also necessary for them to reach their potential as a lingua franca.

Comment:

The workshop has left me feeling one thing very forcefully in connection with this: HCI patterns (perhaps patterns at all) cannot (should not?) be the work of a single person.

Concluding Remarks

Probably the point of greatest agreement among members of the workshop is that we felt that we were at the very beginning of the enterprise of understanding the role and utility of pattern languages for interaction design. Collectively, it is probably fair to say that we are all intrigued by the promise of pattern languages, while, at the same time, a bit wary of expecting too much from what we expect to be just another tool in the interaction designer's repertoire.

Those interested in a greater understanding of patterns and pattern languages should look over the bibliography. For information on the web (and in particular, for information related to software patterns) a good starting point is the Patterns Home Page: <http://st-www.cs.uiuc.edu/users/patterns/>. The Patterns Home Page includes URLs, pointers to mailing lists, and a bibliography. A good beginning for aspiring pattern writers is John Vlissides' essay on pattern making [7], which, though written for the software patterns community, is very useful for those grappling with interaction patterns.

References

1. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S., *A Pattern Language*. New York: Oxford University Press, 1977.
2. Alexander, C., *Timeless Way of Building*. New York: Oxford University Press, 1979.
3. Bohm, D. *On Dialogue*. London: Routledge, 1996.
4. Coplien, J. O., and Schmidt, D. C. (eds.) *Pattern Languages of Program Design*. Reading, Mass: Addison-Wesley, 1995.
5. Gabriel, R. P. *Patterns of Software: Tales from the Software Community*. New York: Oxford University Press, 1996.
6. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass: Addison-Wesley, 1995.
7. Vlissides, J. Pattern Hatching: Seven Habits of Successful Pattern Writers. *C++ Report*, November/December

1996 <http://st-www.cs.uiuc.edu/users/patterns/papers/7habits.html>

A Bibliography of Patterns-Related Material

Here's a list (with occasional annotations) of papers, magazines, and books that were recommended (in various contexts) during the course of the workshop.

Books by Christopher Alexander and his Colleagues

Here is the current corpus of Alexander's books on patterns. His next work, a three volume set, collectively titled *The Nature of Order*, is expected 'shortly' from Oxford University Press.

Alexander, C., *Notes On the Synthesis of Form*. Cambridge, MA: Harvard University Press, 1964.

Alexander's first book, in which he lays out the basic concept of patterns (which he refers to as diagrams).

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S., *A Pattern Language*. New York: Oxford University Press, 1977.

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

Alexander, C., *The Linz Cafe*. New York: Oxford University Press: 1982.

Alexander, C., *The Production of Houses*. New York: Oxford University Press: 1984.

Alexander, C., Neis, H., Anninus, A., King, I., *A New Theory of Urban Design*. New York: Oxford University Press: 1987.

Alexander, C., *The Oregon Experiment*. New York: Oxford University Press: 1988.

Alexander, C. Manifesto 1991. *Progressive Architecture*, July, 1991. pp. 108-112.

Alexander, C., *A Foreshadowing of 21st Century Art: The Color and Geometry of Very Early Turkish Carpets*. New York: Oxford University Press: 1993.

Generally known as 'the rug book'; in it, Alexander tries to analyze what makes early Turkish carpets especially beautiful.

Alexander, C., Black, G., Tsutsui, M.O., Tsutsui, M. *The Mary Rose Museum*.

New York: Oxford University Press: 1995.

On Applications of Pattern Languages in Architecture

These papers describe applications of pattern languages to particular projects; they range from descriptive and enthusiastic to more critical and reflective accounts.

Canine, C & Frankel, G. E. Building By the Book: Using 'A Pattern Language' to design a house with heart. *Harrow-smith*, vol. 3, March 1987. pp. 51-59.

Carpenter, P. Pattern Planning. *Building Ideas*, vol. 5, Summer, 1987. pp. 104-112.

Coates, G. J. & Seamon, D. Promoting a Foundational Ecology Practically Through Christopher Alexander's Pattern Language: The Example of Meadowcreek. In *Dwelling, Seeing, and Designing*, (ed. David Seamon). State University of New York Press: 1993. pp 331-354.

One of the more detailed accounts of applying the pattern language approach to a particular site.

Davis, H. Individual Houses in Groups: The Pattern Language in a Teaching Studio. *Journal of Architectural Education*, vol. 36, 1982. pp. 14-19.

Fallon, R. and Eddington, D. Working With a 'A Pattern Language.' *Fine Homebuilding*. December 1986/January 1987. pp 51-55.

Fromm, D. & Bosselmann, P. Mexicali Revisited: Seven Years Later. *Places*, vol. 1, no. 4, 1983-84. pp 78-90.

Alexander and his students spent time in Mexicali designing housing using a pattern language. This paper, by one of the students, recounts a visit to the housing project seven years later, and reflects on what happened, and ways in which the project failed to achieve its goals, though not without some small successes as well.

Hester, R. Sacred Structures and Everyday Life: A Return to Manteo, North Carolina. In *Dwelling, Seeing, and Designing*, (ed. David Seamon). State University of New York Press: 1993. pp. 271-297.

Not about use of an Alexandrian pattern language, per se, but about the creation of a pattern language-like

representation which came to serve as a common language in a participatory design process.

Mulfinger, D. Putting 'A Pattern Language' to Work. *Fine Homebuilding*, Spring, 1987. pp. 49-53.

Other Readings in Architecture and Urban Design

While these references are not about the pattern language approach, *per se*, their approaches to architectural and urban design are in much the same spirit.

Bacon, E. N. *Design of Cities*. New York: Penguin Books, 1967.

Banerjee, T. and Southworth, M. *City Sense and City Design: Writings and Projects of Kevin Lynch*. Cambridge: The MIT Press, 1990.

The work of Kevin Lynch is well worth a look by anyone interested in a holistic, experiential approach to urban design. This book is a collection of writings spanning his career; also of interest is the seminal, Image of the City (1960), as well as What Time is this Place (1972), and Good City Form (1981), all from MIT Press.

Bentley, I., Alcock, A., Murrain, P., McGlynn, S., & Smith, G. *Responsive Environments*. London: The Architectural Press, 1985.

Brand, S. *How Buildings Learn: What Happens After They're Built*. New York, NY: Penguin Books Inc., 1994.

Cuff, D. *Architecture: The Story of Practice*. MIT Press. 1991

Garreau, J. *Edge City*. New York, NY: Anchor Books Inc., 1991

Gehl, Jan. *Life Between Buildings: Using Public Space*. New York: Van Nostrand Reinhold, 1980/1987.

Jacobs, J. *The Death and Life of Great American Cities*, New York: Random House, 1961.

Arguably one of the century's most significant and influential books on urban design.

Seamon, D. (editor). *Environmental and Architectural Phenomenology Newsletter*. Kansas State.

A small circulation, quarterly newsletter on phenomenological approaches to architecture and urban design. (Subscription: \$10 (\$12 non-USA)/year payable to David Seamon/EAP, at

David Seamon; *Architecture Dept.*; 211 Seaton Hall, Kansas State University; Manhattan, KS 66506. 913-532-5953.)

Seamon, D. (editor) *Dwelling, Seeing, Designing: Toward a Phenomenological Ecology*. (SUNY Series in Environmental and Architectural Phenomenology). State Univ of New York: February, 1993.

Whyte, W.H. *City: Return to the Center*. New York: Anchor Books, 1988.

A sociologist reports on a sixteen year study of social behavior in urban spaces.

Patterns for Software and Information Systems

Coplien, J. O. *Patterns*. SIGS Books, 1996.

Coplien, J. O., and Schmidt, D. C. (eds) *Pattern Languages of Program Design*. Reading, Mass: Addison-Wesley, 1995.

Curtis, B., Krasner, H., & Iscoe, N. *A Field Study of the Software Design Process for Large Systems*. Communications of the ACM 1988;31(11):1268-1287.

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

This book touches on some of the overlaps between Alexandrian pattern languages, and patterns as they apply to software design.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass: Addison-Wesley, 1995.

Often referred to, in the software patterns community, as the 'Gang of Four book.'

Lea, D. *Concurrent Programming in Java: Design Principles and Patterns*. Addison-Wesley, 1996.

Markus M.L., & Keil M. *If We Build It, They Will Come: Designing Information Systems That Users Want To Use*. *Sloan Management Review*, Summer 1993. pp11-25.

Schmidt, D., Fayad, M., & Johnson, R. (eds.). Special Section on Software Patterns. *Communications of the ACM*, vol. 39, no. 10, October 1996. pp. 36-86.

Vlissides, J. *Pattern Hatching: Seven Habits of Successful Pattern Writers*. *C++ Report*, November/December 1996. <http://st-www.cs.uiuc.edu/users/patterns/papers/7habits.html>

Vlissides, J. *Patterns: The Top Ten Misconceptions*. http://www.sigs.com/publications/docs/objm/9703/9703_vlissides.html

Organization Design

Collins, J.C., & Porras, J.I. *Built to Last: Successful Habits of Visionary Companies*. Harper-Collins 1997.

Senge, P. *The Fifth Discipline: The Art and Practice of a Learning Organization*. Doubleday 1994.

The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization. Peter M. Senge, Charlotte Roberts, Richard B. Ross (eds.) Doubleday: 1994.

Other

Glaser, B.G., & Strauss A.L. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Hawthorne, N. Y.: Aldine de Gruyter, 1967.

Zuboff, S. *In The Age of The Smart Machine: The Future of Work and Power*. New York, NY: Basic Books Inc., 1988.

The Workshop Organizers

Thomas Erickson is an interaction designer at IBM's T.J. Watson Labs (he was at Apple Research at the time of the workshop). His current research interests include social interaction over the internet, digital genres, personal information systems, and interaction design theory and practice. He can be contacted at snowfall@acm.org, and browsed at http://www.pliant.org/personal/Tom_Erickson/

John Thomas is an Executive Director for Bell Atlantic Science and Technol-

ogy with responsibilities for speech technology, applications, and platforms as well as human computer interaction. He has published numerous articles in such areas as speech technology, query languages, human problem solving and creativity, and human computer interaction. Currently, he is focusing his own work on Organizational Learning. He can be contacted at thomas@nynexst.com and browsed at <http://www.truthtable.com>.

The Workshop Participants

Elisabeth Bayle, Bayle Collaborations
Rachel Bellamy, Apple Computer
George Casaday, Marcam Corporation
Thomas Erickson, Apple Computer
Sally Fincher, University of Kent at Canterbury
Beki Grinter, Lucent Technologies
Ben Gross, University of Illinois at Urbana-Champaign
Diane Lehder, AT&T Labs
Hans Marmolin, UI Design
Brian Moore, Texas Instruments
Colin Potts, Georgia Tech
Grant Skousen, Consultant
John Thomas, NYNEX

Contact

Thomas Erickson
IBM Research, T.J. Watson Labs
P.O.Box 704, H1-M09
Yorktown Heights, NY 10598
snowfall@acm.org
http://www.pliant.org/personal/Tom_Erickson

John C. Thomas
Bell Atlantic Science and Technology
500 Westchester Avenue
White Plains, NY 10604
thomas@nynexst.com
<http://www.truthtable.com>

Editor's Note: for an earlier SIGCHI Bulletin article on the use of Pattern Languages in Interaction Design, see "The Timeless Way... The Design of Meaning" by Dick Rijken et al., SIGCHI Bulletin, Vol. 6, No. 3, July 1994, pp. 70-79.