Patterns: What's In It For HCI?

Jan O. Borchers
Stanford University
Stanford, CA 94305-9020, USA
+1 650 725 7646
borchers@stanford.edu

John C. Thomas
IBM Research, P. O. Box 704
Yorktown Heights, NY 10598, USA
+1 914 784 7561
jcthomas@us.ibm.com

ABSTRACT

Design patterns and pattern languages have proven a useful tool to model design experience, in architecture where they were originally conceived as well as in software engineering. In Human-Computer Interaction, the interest in pattern languages has only recently gained momentum. This panel will explore how pattern languages can be of use to HCI researchers, practitioners, and possibly anyone involved in the design and use of interactive systems.

Keywords

pattern languages, design patterns, design methodologies

PANELISTS

- **John Thomas**, IBM Research (moderator)
- **Alistair Sutcliffe**, UMIST
- **Jim Coplien**, Bell Labs
- **Richard N. Griffiths**, University of Brighton
- **Tom Erickson**, IBM Research
- **Jan Borchers**, Stanford University

INTRODUCTION

Design patterns describe successful solutions to recurring design problems, and are organized hierarchically into a *pattern language*. Their form has proven to be a useful tool to model design experience, in architecture where they were originally conceived [1, 2] as well as in software engineering which picked up the concept in the late eighties and since established an active community [5]. In Human-Computer Interaction (HCI), although early references exist [9], it is only recently that the interest in pattern languages has gained momentum, as indicated by a series of workshops [3, 7, 8] and publications.

The INTERACT'99 workshop [7] defined the goals of an HCI pattern language to be " . . . to share successful HCI design solutions among our colleagues, and to provide a common language for HCI design to anyone involved in the design, development, evaluation or use of interactive systems."

PANEL FORMAT

Since the audience may not be familiar with the idea of HCI design patterns, we will begin with a concise summary of what they are about, gather some opinions from the audience, and hand out copies of HCI design patterns for illustration. To convey the variety of viewpoints represented by the panelists, they will outline why they are interested in HCI design patterns, and give an example of how they recently used them in their work. We will then discuss the following fundamental and controversial questions about HCI design patterns, to outline where this field stands today:

1. Patterns are en vogue in software engineering, but were a notable non-success in their original field of architecture. Why should they be successful in HCI?
2. Who are HCI design patterns for, and what should they be used for—as Design Rationale, to replace standards or guidelines, or for training in industry and academia?
3. What kind of knowledge should HCI design patterns capture, and at which level of abstraction? Should they be integrated with software engineering patterns, and how?
4. How can we find HCI design patterns, and evaluate them?
5. To what extent do pattern languages predispose the investigator to see only what the language makes expressible?
6. In what form should HCI design patterns be presented: formal or informal; as UML or narrative?
7. Should we try to construct 'permanent' HCI patterns and languages so that they can be reused, or is the process of seeing and talking in terms of patterns the primary aim?
8. Should patterns be owned by individuals or institutions, or should they be collective, 'open source' products?

The remainder of the panel will let the audience judge the patterns handed out in terms of perceived usefulness for their work, and we will ask for further questions and comments.

POSITION STATEMENTS

All participants are key players in the emerging field of HCI design patterns, and have been selected to span a wide variety of different backgrounds and approaches to the subject.

John Thomas: Codifications of knowledge such as "cookbooks", "software packages", "quantitative trend analyses" and a "standard set of GUI widgets" are not the appropriate level of generality in a field such as Human-Computer Interaction today, where changes are rapid and pervasive. Pattern Languages help a community of practitioners such as HCI experts develop and share knowledge over time and circumstance at an appropriate level of generality and flexibility. For example, we focus on socio-technical patterns in the development of on-line communities. These offer new challenges for dealing with trust, accountability, and informed consent. While the instantiation of socio-technical patterns into specific solutions will change over time, the general patterns will remain useful as tools of thought. For example, communities need to learn from an honest examination of failure without censuring individuals. Anonymized stories, cautionary tales, and "Mullah Rabin" stories are patterns which societies have developed in a proto-technological world which are still applicable in a wired world.

Alistair Sutcliffe: The key problem for patterns is to develop a sound theory of abstraction. The essence of patterns is their shareability. To be shared they have to be understood by others and composable so different patterns will fit together in new solutions. HCI patterns may slowly converge into a standand framework via writers' workshops and multiple, solo authored pattern languages, but progress will be slow and painful. Some serious research is required to develop a framework of sharable knowledge so pattern authors can target their contributions to different concerns in HCI, such as task support, interaction design, information and presentation.

Jim Coplien: The fact that patterns have a technical basis in geometry, and an ethnographic basis in community and human comfort, is largely lost on contemporary pattern literature. The obvious link from the HCI community to geometry, through its concern with graphical interfaces, and its direct link to human usability, offer hope that the field of human-computer interaction may provide the most fertile soil for the pattern vision to come to fruition.

Richard N. Griffiths: Casting good HCI design practice into the form of patterns may be advantageous in situations where guidelines may have previously been used; but the immaturity of HCI design and the continuing rapid development of its supporting medium (display technology, memory capacity, processor speed, etc.) make the quality of patterns produced dubious, and their applicability ephemeral. However, the idea of patterns, particularly with reference to Alexander's earlier work on design space analysis, has profound implications for HCI design practice. It emphasises the capture and analysis of the full range of requirements—technical, economic, physiological, psychological and social (no significance is implied by the sequence)—that must be addressed in the implementation of each design detail, and proposes a method whereby this otherwise overwhelming complexity can be handled. Thus in the short term, it is thinking in patterns when designing HCI, rather than the production of patterns per se that is appropriate.

Thomas Erickson: I am interested in pattern languages as lingua francas, common languages that can facilitate egalitarian communication amongst all stakeholders in a design project [6]. Many advocates of patterns seem to ignore the generative nature of pattern languages. That is, patterns are often viewed as a sort of poetic form of guidelines. However, in my view, and in the original conception of pattern languages, they are intended to be meta-languages which are then used to generate project-specific languages that are grounded in the social and cultural particularities of a given design domain. I view pattern languages as a way of allowing people to see and talk about the interdependencies in a particular design domain. I am not convinced of the value of using patterns prescriptively (because I am not convinced that interaction design has sufficient experience to be confident of its solutions), nor do I see a monolithic or general pattern language as necessary or desirable.

Jan Borchers: HCI is much closer to architecture than to software engineering. Therefore, we should largely adapt Alexander's notion of patterns, and change it consciously for HCI. Just using patterns as a format for expert communication, as it is done in current software engineering pattern practice, ignores their potential. Instead, I suggest using them to express design experience not only in HCI and software engineering, but also in the application domain of interactive system design projects, and to make sure they are readable by professionals and non-professionals alike. This creates three pattern languages that help the interdisciplinary design team talk to each other [4].

REFERENCES

[1] C. Alexander. *The Timeless Way of Building*. Oxford University Press, 1979.

[2] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

[3] E. Bayle, R. Bellamy, G. Casaday, T. Erickson, S. Fincher, B. Grinter, B. Gross, D. Lehder, H. Marmolin, B. Moore, C. Potts, G. Skousen, and J. Thomas. Putting it all together: Towards a pattern language for interaction design. *SIGCHI Bulletin*, 30(1):17–23, Jan. 1998.

[4] J. Borchers. *A Pattern Approach to Interaction Design*. 260 pages. John Wiley & Sons, February 2001.

[5] J. O. Coplien and D. C. Schmidt. *Pattern Languages of Program Design*. Software Patterns Series. Addison-Wesley, 1995.

[6] T. D. Erickson. Lingua Francas for Design: Sacred Places and Pattern Languages. *Proc. DIS 2000 (New York, Aug. 17–19, 2000)*, pp. 357–368. ACM Pr., 2000.

[7] R. N. Griffiths, L. Pemberton, and J. O. Borchers. Usability pattern language: Creating a community. In S. Brewster, A. Cawsey, and G. Cockton, eds., *Human-Computer Interaction: INTERACT'99, Vol. II*, p. 135. The British Computer Society, 1999.

[8] R. Griffiths, L. Pemberton, J. Borchers, and A. Stork. Pattern languages for interaction design: Building momentum. *CHI 2000 Extended Abstracts (The Hague, Apr. 1–6, 2000)*, p. 363. ACM Press, 2000.

[9] D. A. Norman and S. W. Draper. *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, 1986.