Panel: Patterns in Human-Computer Interaction Design

Richard N Griffiths
University of Brighton
School of Computing and Mathematical Sciences
Brighton, BN2 4GJ
r.n.griffiths@bton.ac.uk

Lyn Pemberton
University of Brighton
School of Information Management
Brighton, BN2 4GJ
LP22@bton.ac.uk

SUMMARY

Design patterns and pattern language has recently emerged as an approach to human-computer interaction design. This follows their adoption by the software engineering community. In many respects HCI being closer to urban and building design—the domain in which pattern language was developed—than software architecture, may have more to gain. This panel will bring together a number of people involved with patterns in HCI to present these ideas, and air controversies that surround them.

KEYWORDS: Patterns, pattern language, user-centred, design, HCI.

INTRODUCTION

The idea of recording successful solutions to recurring design problems as *design patterns* was taken up by the software engineering community in the late 1980's [5, 9, 13, 8]. The immediate precursor to this use was the development of a particular approach to urban and building design by the architect Christopher Alexander and colleagues in the 1970's [1, 2]. It is perhaps surprising that an idea developed to express design solutions that directly influence the occupants' experience of architecture, should first be taken up within the software domain by designers of software architecture—where there is no necessary impact upon the users. Actually, some significant early developments in human-computer interaction (HCI) design were influenced by the design pattern idea [16, 15, 3]. However interest in the development of specifically HCI design patterns is relatively recent, stimulated by a series of international workshops [4, 10, 11, 6, 12] and both formal and informal publications [14]. A Task Group on HCI Design Patterns has been set up under IFIP Working Group 13.2: Methodology for User Centred Design, to promote this development.

The community's understanding of the design pattern concept and its application in HCI design has undergone some evolution throughout this initial period. At first sight, the idea of a design pattern appears perfectly straightforward: it is just a particular format for documenting extended guidelines. The *pattern language* in which design patterns are embedded is simply an organisational device (hierarchical decomposition) to facilitate the identification of a required pattern.

This first impression is however deeply misleading. The form of design patterns as presented by Alexander, and adopted by many pattern writers in different fields, was intended merely as a vehicle for the transmission of a deep, 'user centred' philosophy of design. Essentially, for Alexander, a pattern is a fundamental invariant in the deep geometry of the world. (Thanks to Tom Ericson for that phrase.) The social and personal dimensions of human experience are part of this multidimensional space. A documented design pattern attempts to convey the essence of an invariant, in a useful way so that it may be incorporated into new designs.

Proponents (and detractors) of pattern languages, acknowledge this philosophical underpinning to a greater or lesser extent. In fact, some proponents who use design patterns to practical effect find reference to Alexander's intention for them deeply irritating and irrelevant. Others would see this as entirely missing the point. This in turn feeds a dichotomy in approach to using patterns between those who emphasise the product (published design patterns) and those who emphasise the process (the development and use of patterns to enhance communication between the stakeholders in a project). There may well be merit in both of these positions, and others yet to be articulated. This will be determined through experience with patterns in HCI design, which is starting to accumulate, and debate amongst proponents, which initiative such as this panel carry forward.

FORMAT

We propose to adopt the format that proved very successful in the panel discussion on patterns at CHI 200l. The different positions came across well, and lively participation from the audience was provoked.

As the concept of design patterns in HCI may be new to many of the audience, it will be necessary to begin with a brief introduction and background. This should take no longer than 15 minutes. Examples of HCI patterns and their application will be given.

Each panellist will be given one minute to present a succinct statement of their position.

Discussion will be initiated by the moderator posing a broad question on the status of patterns in HCI design, intended to bring out differences between the panellist. After responses from the panel, questions and comments will be invited from the audience for the panel's response. Based on prior experience we envisage this will generate lively debate within the panel, and with the audience. The moderator will have a reserve list of questions should audience participation fail to materialise.

POSITION STATEMENTS FROM EACH PANELLIST

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.

Richard Griffiths: The immaturity of HCI design and the continuing rapid development of its supporting medium (display technology, memory capacity, processor speed, etc.) makes the quality of patterns produced dubious, and their applicability ephemeral. However, the *idea* of patterns, if understood in depth, and particularly with reference to Alexander’s work on design space analysis which preceded the development of pattern language, has profound implications for HCI design practice. It places particular emphasis on the capture and analysis of the full range of requirements; technical, economic, physiological, psychological and social 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.

Martin Hitz: Usability / HCI Patterns are a very exciting idea. But it remains to be seen whether they will have as successful a story as patterns in the area of software engineering. The reasons for a possible decline of the idea being: HCI seems to be quite congenial to Architecture, an area where Alexander's approach has failed (in my opinion). There seems to be a good deal of an 'artist' in the self-image of 'interface architects' which might reduce the willingness to adhere to somebody else's predefined solution patterns. We are still missing a standard 'bible' in our field, something that spreads the idea, especially among the practitioners like the GOF book. In order to create such reference work, we need a common understanding of the pattern languages to be used and the different subfields ('strata') of HCI to which patterns should apply. In software engineering, part of the success seems to be rooted in the fact that already very early different strata evolved which gave rise to relatively homogenous treatment of DESIGN patterns, ANALYSIS patterns ARCHITECTURAL patterns.

Lyn Pemberton: There is a long list of interesting issues to be pursued at this stage in the application of design patterns to interaction design. The most immediate for me are those relating to the use of patterns in HCI teaching, where they can play two main roles:

1. As a knowledge representation formalism: patterns can be an alternative/complement to guidelines as a way of packaging up craft knowledge
2. As a thinking tool: abstracting away from concrete examples to formulate general patterns is potentially a way of ensuring that students have a deep understandomg of interaction problems and can find new, creative solutions outside current paradigms.

Helen Sharp: My position is based on my experience of co-leading the Pedagogical Patterns Project which started in 1996, rather than any experience of trying to apply pattern ideas in HCI. When the project started, we didn’t intend to collect patterns, we simply wanted to collect together reusable experience of teaching Object Technology in a homogeneous form that could be shared easily. In this sense, patterns have helped us achieve our goals, and have produced a community of people around the world who have all contributed their ideas and comments for others to see, use, build upon and learn from.

BIOGRAPHICAL SKETCHES OF PARTICIPANTS

Jan Borchers: Jan received his Ph.D in Computer Science from Darmstadt University of Technology and an M.Sc. in Computer Science from the University of Karlruhe. He is currently an Assistant Professor in the Computer Science Department at Stanford University. Since 1995, he has designed and managed the development of a series of interactive exhibits, and has been Visiting Scientist and lecturer at the University of Ulm in Germany since 1996. He participated in the first workshop on interaction patterns with the software engineering patterns community [6], and co-rganized the workshops on pattern languages in HCI at INTERACT’99 and CHI2000, and the pannel at CHI2001. He has published various articles on this and related subjects. He has also used HCI design patterns in his user interface design courses, and he is the authorof the first book about HCI patterns [7].

Richard Griffiths: Richard has a first degree in Sociology and an MA in Cognitive Studies. He has worked in commercial application development for John Hoskyns & Co. and system software development for Racal before becoming a lecturer and researcher in computing. He now specialises in human-computer interaction in the School of Computing and Mathematical Sciences at the University of Brighton, UK. He is currently collaborating with Victoria Real Ltd., a leading UK digital media company, in the development of a post-graduate course for interactive television design. He has co-lead workshops on patterns in interaction design at INTERACT'99 and CHI 2000, and was a panellist in a session entitled 'Patterns: What's in it for HCI?' at CHI'01.

Martin Hitz: Martin is a full professor of Interactive Systems with the Institute for Computer Science-Systems at the University of Klagenfurt. Before joining the institute he was associate professor at the University of Vienna. He earned a MSc in computer science in 1982 and a PhD in computer science in 1989 from the Vienna University of Technology, respectively. Research interests include software engineering, HCI, multimedia systems, and web-based information systems.

Lyn Pemberton: Lyn's academic background is originally in Language and Literature. After an MSc in Knowledge Based Systems she worked at the University of Sussex on educational software. Since 1990 she has been engaged in researching, teaching and consulting in human-computer interaction at the University of Brighton, UK. She has recently completed an EU-funded project on language learning software for children who speak minority languages. Lyn has been involved in workshops on patterns in interaction design at INTERACT'99 and CHI 2000.

Helen Sharp: Helen Sharp in a Senior Lecturer in the Centre for HCI Design at City University, London. She co-led the Pedagogical Patterns Project from the beginning of 1996 to the end of 2000, when the original Project leaders handed over the reins to others who continue the work. The PPP is an unfunded international project seeking to build a pattern language or languages for teaching and learning Object Technology. Helen is widely-known in the Object Technology world, and is also the co-author of a leading HCI textbook. With Jenny Preece and Yvonne Rogers she has recently produced another HCI textbook due out later in 2001.

BIBLIOGRAPHY

1. Alexander, C. *The Timeless Way of Building*. Oxford University Press, 1979.
2. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.
3. Apple Computer. *Macintosh Human Interface Guidelines*. Addison-Wesley, 1992.
4. Bayle, E., et. al. Putting it all together: Towards a pattern language for interaction design. *SIGCHI Bulletin*, 30(1):17–23, January 1998.
5. Beck, K. and Cunningham, W. Using pattern languages for object-oriented programs. Technical Report CR-87-43, Tektronix, Inc., September 17, 1987. Presented at the *OOPSLA'87 workshop on Specification and Design for Object-Oriented Programming.*
6. Borchers, J.O. CHI meets PLoP: An interaction patterns workshop. *SIGCHI Bulletin*, 32(1):9–12, January 2000.
7. Borchers J.O. *A Pattern Approach to Interaction Design*. John Wiley & Sons, 2000.
8. Coplien J.O. and Schmidt D. C. *Pattern Languages of Program Design*. Software Patterns Series. Addison-Wesley, 1995.
9. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
10. Granlund, A. and Lafrenère, D. A Pattern Supported Approach to the User Interface Design Process, workshop at the *Usability Professionals Association Annual Conference,* Scottsdale, AZ, June 29 – July 2, 1999.
11. Griffiths, R.N., Pemberton, L. and Borchers, J.O. Usability pattern language: Creating a community. In S. Brewster, A. Cawsey, and G. Cockton, editors, *Human-Computer Interaction - INTERACT'99 (Volume II)*, page 135. The British Computer Society, Wiltshire, UK, 1999.
12. Griffiths, R.N., Pemberton, L., Borchers, J.O. and Stork, A. Pattern languages for interaction design: Building momentum. In Szwillus, G. and Turner, T. editors, *CHI 2000 Extended Abstracts (Conference on Human Factors in Computing Systems, The Hague, Netherlands, April 2–3,2000)*, page 363. ACM Press, New York, 2000.
13. Harrison, N., Foote, B. and Rohnert, H. editors. *Pattern Languages of Program Design* 4. Software Patterns Series. Addison-Wesley, 1999.
14. hcipatterns.org: The HCI Patterns Pages. Accessible at http://www.stanford.edu/~borchers/hcipatterns/
15. Norman, D.A. *The Psychology of Everyday Things*. Basic Books, New York, 1988.
16. Norman, D.A and Draper, S.W. *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.