Identifying Promising Objectives for a Sustainable HCI Pattern Language

Christian Remy
University of Zurich
Binzmühlestrasse 14
8046 Zurich, Switzerland
remy@ifi.uzh.ch

Elaine M. Huang
University of Zurich
Binzmühlestrasse 14
8046 Zurich, Switzerland
huang@ifi.uzh.ch

Abstract
Sustainable HCI can still be considered a relatively young field; yet, it has seen a variety of recent contributions investigating previous approaches and raising questions on how to go forward. As such, the field would greatly benefit from an effort to summarize existing solutions, avoiding reinventing the wheel but also identifying blank spots of missing research. Design patterns offer such an approach. Established in the domain of architecture, successfully utilized to date in software engineering, and having been applied to a variety of areas in HCI, design patterns have a rich history from which the SHCI community can learn and use it to its advantage. We examine previous approaches of design patterns and based on those insights lay out a set of challenges and opportunities for their application to SHCI. As our analysis highlights, patterns can be geared towards recording knowledge or as a tool for outward communication, and given the state of the field, we believe the first objective is the more feasible and more promising starting point.

Author Keywords
Design Patterns, Pattern Language, Sustainable HCI

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

Introduction

The field of sustainable HCI has become an established domain of research within the HCI community, and a significant number of contributions have sought to bring the field forward in recent years. While the amount of publications grows and the research agenda evolves, it becomes more and more difficult to keep track of where we came from and where we are heading in this field. Early surveys [5,8] already highlighted the diversity of the field six years ago, which has increased significantly since. Especially for new researchers coming into this field, the plethora of work can easily be overwhelming. One prime example is how the discussion surrounding eco-feedback technology has evolved: from early enthusiasm and spurred by a surge of different approaches [8,13,17] the discussion shifted to how to actual achieve the goal of behavior change, and yielded nuanced pieces of advice on how to go forward [3,11,13,16]. When looking to identify the next steps, it is equally important to look back and as objectively as possible summarize what the achievements as well as challenges were – and what challenges and opportunities lie ahead derived from these insights.

We argue that a pattern language [1] could be one way to document all those past research efforts and create a well-structured, easily accessible source of sustainable HCI knowledge. A design 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" [1, p. x]. In other words, a pattern describes established knowledge in a field, but also leaves room for creativity when creating new solutions for an already known problem – without reinventing the wheel. Since research is constantly evolving due to new insights and novel approaches that are constantly being developed, such a pattern language should be flexible and not set in stone. Therefore, we believe that a community effort to develop, maintain, and revisit patterns would prove to be beneficial for the field. In the following, we will briefly revisit the history of design patterns and highlight lessons learned from past efforts. Based on this, we will outline opportunities and challenges that arise for the field of sustainable HCI, in particular with respect to aspects unique to the field. A proposal for how to proceed and how to maximize success of this approach will conclude our submission.

Success Stories of Design Patterns

In the 1960s and 1970s, Architect Christopher Alexander sought new ways to bring forward the domain of architecture, but also empower nonprofessionals to be able to engage into discussions with domain experts during the planning process of architecture projects. One of these approaches was *A Pattern Language* [1]: a seminal and highly regarded book that articulated standards of architectural design. The design patterns were made easy to understand by avoiding genre-specific jargon, the language was easy to browse due to links between the patterns, and illustrated with examples, pictures, and short statements summarizing the core essence. Other disciplines quickly picked up the concept, but most notably design patterns made an impact on the software engineering community, in particular with the seminal book *Design Patterns* [10]. In the domain of HCI, the early 2000s saw a surge of HCI design pattern proposals as well [e.g., 2,5,12].

Our next step is to critically examine how significant the impact of design patterns in their respective disciplines was. In architecture, Alexander's concepts found widespread critical acclaim – but were also met with controversy [4, p. 56]. On the contrary, design patterns became well-established as an alternative form of guidelines in software engineering. Following the success of Gamma et al.'s book [10], the *Pattern Languages of Programming*[1] *(PLoP)* is an annual conference that creates, debates, and publishes design patterns, and its organizing body, The Hillside Group, currently lists 81 books on their website[2]. One important difference between the original design patterns and their adaption in software engineering is that the latter were not meant to cross disciplines or make software engineering solutions understandable to nonprofessionals. Rather, Gamma et al. [10] focused on documenting solutions to recurring problems, creating a body of knowledge from experts for experts.

HCI design patterns lie somewhere in between: As an interdisciplinary field, prospective readers oftentimes are expected to not be experts in a certain problem domain. Thus, several pattern languages have adhered to Alexander's style [e.g., 2,11] or were phrased in a way to be easy to understand for novice readers [e.g., 6,14], emphasizing Erickson's call for patterns as a lingua franca [7]. Dearden and Finlay's analysis [4] categorizes five use cases of patterns: lingua franca and participatory design, which can be attributed to efforts of engaging non-experts through design patterns; as well as technical lexicon, organizational memory, and design rationale, which are more about creating a resource from experts for experts.

A Sustainable HCI Pattern Language

Pan and Stolterman state that a pattern language "is only a method for knowledge representation, not a research process method" and, as several senior HCI researchers state in their interview study, a pattern language is "not a prescriptive tool for design" [12]. Their principle argument is that pattern languages take a lot of time and effort to be created – a notion that is mentioned by a variety of design pattern experts, including Alexander himself [1], whose 253 patterns were eight years in the making. Once such a comprehensive language is conceived, "it requires users (designers) who spends a lot of time to become familiar with the set of patterns" [12]. This aligns with observations of recent pattern languages – including the popular and successful examples from software engineering – focusing more on the aspects of creating a knowledge repository for the discipline itself rather than practitioners outside of the field, as referred to in the call for a lingua franca [7]. Furthermore, patterns in computer science change over time since the discipline is highly dynamic, especially in a relatively young field such as SHCI. All those insights point to focus on creating an organizational memory first rather than considering patterns as a tool for communicating knowledge.

From these past experiences with design patterns in HCI and other domains, as well as previous critiques and investigations of design patterns, we lay out a set of challenges and opportunities for SHCI:

[1] http://www.hillside.net/plop/

[2] http://hillside.net/patterns/books/design-patterns

A Pattern Language from SHCI for SHCI
As investigations of existing pattern languages highlighted, the most successful approaches saw pattern languages being used as a tool to document knowledge in a certain domain for other domain experts or people who sought to become experts. This would also allow the community to identify and agree upon existing solutions in the field as well as document those for new researchers entering the field. A translation into a lingua franca for practitioners outside the field can be discussed once the pattern language has reached a certain level of maturity – although even then it faces the issue that "the expertise and skill needed by those who develop a PL is also needed by those who attempt to use it" [12]. This is even more so the case for the field of SHCI where discussions about how to bring the field forward surface frequently [e.g., 3,5,10,12,14] and the research agenda is in constant change.

A Pattern Language rather than a Pattern Collection
Previous investigations [4,12] highlighted the difference between a pattern language and a collection of patterns. Especially when developing design patterns, considering an ecology of connected patterns helps to identify blank spots of missing topics, but also has the advantage of creating a hierarchy in terms of the patterns' importance and applicability. Even the first steps of generating initial design patterns might provide SHCI with answers to the question of what the "next steps for SHCI" are [16]. In addition, pattern languages as opposed to pattern collections have the advantage of being generative [1,4], which increases their usefulness when applying them to practice – an aspect in which SHCI research is lacking [5,14].

Continuously Evaluating Design Patterns
A concern that is often raised about assessing the quality of a pattern is its evaluation. HCI pattern languages are often evaluated by applying them to practice; however, many of those evaluations are not empirical or objective enough to assess the real effectiveness [4,12]. An established method is the writers' workshop [9] which is used to evaluate pattern submissions at PLoP conferences, a format similar to a focus group, although with stricter guidelines. Such a format has limitations – it requires either physical presence of a number of people including the authors, or at least a virtual conference. Furthermore, the patterns need to be at a certain level of maturity rather than just be "pattern stubs" or "proto patterns". It is also beneficial to have a relatively small group to still be constructive and feasible. We believe such an evaluation could be applied once the patterns are more developed, but for the initial approach a more open platform as well as an inclusive process that does not limit the number of potential participants might be more effective. Because SHCI is still a relatively new field and solutions are dynamic, any evaluation process should not be a one-time review but an effort to combine several different evaluation methods into an iterative process.

Maintaining a Pattern Language Repository
The most difficult challenge probably is not to stop halfway through the process, i.e., after a few "pattern stubs" have been developed at the workshop, but keep up the momentum and community engagement. There are several existing examples of online pattern repositories; the websites of Tom Erickson[3] and Jan

[3] http://www.tomeri.org/InteractionPatterns.html

Borchers[4] provide links to some of those. Most of those pattern languages face the challenge of being quite general, as the field of interaction design has grown to an extent where a single pattern language can hardly capture all existing solutions. In contrast, the design space of existing contributions within SHCI is still significantly smaller and thus the community has a unique chance to create a comprehensive pattern language at this point. Unlike many other workshops, SHCI has proven that it is capable of creating a community-generated document based on a collective effort [16].

Conclusion

Design patterns have a rich history in HCI as well as other domains from which we believe the SHCI community can learn on how to adapt it best to its needs. We argue that establishing a pattern language that categorizes established solutions in the field of SHCI not only helps to streamline discussions and achieve agreement on how to counter recurring problems; it also offers the possibility of identifying blank spots and answering the question of how to go forward. This workshop offers a unique opportunity to create the foundation for such a pattern repository, and although a variety of challenges lies ahead, it is a worthwhile effort to create a dynamic knowledge repository for SHCI researchers.

References

1. Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.
2. Jan Borchers. 2001. *A Pattern Approach to Interaction Design*. Wiley, Chichester, England ; New York.
3. Hronn Brynjarsdottir, Maria Håkansson, James Pierce, Eric Baumer, Carl DiSalvo, and Phoebe Sengers. 2012. Sustainably unpersuaded: how persuasion narrows our vision of sustainability. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ACM, 947–956.
4. Andy Dearden and Janet Finlay. 2006. Pattern Languages in HCI: A Critical Review. *Human–Computer Interaction* 21, 1: 49–102.
5. Carl DiSalvo, Phoebe Sengers, and Hrönn Brynjarsdóttir. 2010. Mapping the landscape of sustainable HCI. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1975–1984.
6. Douglas K. van Duyne, James A. Landay, and Jason I. Hong. 2006. *The Design of Sites: Patterns for Creating Winning Web Sites*. Prentice Hall, Upper Saddle River, NJ.
7. Thomas Erickson. 2000. Lingua Francas for Design: Sacred Places and Pattern Languages. *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, ACM, 357–368.
8. Jon Froehlich, Leah Findlater, and James Landay. 2010. The design of eco-feedback technology. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1999–2008.
9. Richard P. Gabriel. 2002. *Writers' Workshops & the Work of Making Things: Patterns, Poetry...* Pearson Education, Boston.
10. Erich Gamma, Richard Helm, Ralph Johnson, and Grady Booch. 1994. *Design Patterns: Elements of*

[4] http://www.hcipatterns.org/patterns

Reusable Object-Oriented Software. Addison-Wesley Professional, Reading, Mass.

11. Bran Knowles, Lynne Blair, Stuart Walker, Paul Coulton, Lisa Thomas, and Louise Mullagh. 2014. Patterns of Persuasion for Sustainability. *Proceedings of the 2014 Conference on Designing Interactive Systems*, ACM, 1035–1044.
12. Yue Pan and Erik Stolterman. 2013. Pattern Language and HCI: Expectations and Experiences. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ACM, 1989–1998.
13. James Pierce and Eric Paulos. 2012. Beyond energy monitors: interaction, energy, and emerging energy systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 665–674.
14. Christian Remy, Silke Gegenbauer, and Elaine M. Huang. 2015. Bridging the Theory-Practice Gap: Lessons and Challenges of Applying the Attachment Framework for Sustainable HCI Design. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 1305–1314.
15. Christian Remy, Malte Weiss, Martina Ziefle, and Jan Borchers. 2010. A Pattern Language for Interactive Tabletops in Collaborative Workspaces. *Proceedings of the 15th European Conference on Pattern Languages of Programs*, ACM, 9:1–9:48.
16. M. Six Silberman, Lisa Nathan, Bran Knowles, et al. 2014. Next steps for sustainable HCI. *interactions* 21, 5: 66–69.
17. Yolande A.A. Strengers. 2011. Designing Eco-feedback Systems for Everyday Life. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2135–2144.
18. Jenifer Tidwell. 2005. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, Beijing ; Sebastopol, CA.