Teaching HCI Design Patterns: Experience From Two University Courses

Jan Borchers
Stanford University
borchers@cs.stanford.edu

ABSTRACT

This paper summarizes the findings of teaching and evaluating two courses within computer science curricula that dealt with HCI Design Patterns.

Introduction

An HCI Design Pattern captures the essence of a successful solution to a recurring usability problem in interactive systems. It consists of a name, ranking, sensitizing example, context, problem statement, evidence (rationale, examples), solution, sketch, references to other patterns, synopsis, and credits (CHI 2000 workshop definition, reported in [1, p. 179ff]).

HCI Design Patterns have a variety of uses, including informing the current user interface design team within a project, capturing best practice for follow-up projects and the general profession, and teaching basic guidelines for HCI design to newcomers.

I have used the pattern format to teach HCI basics to computer science students within two quite different courses. The rest of this paper briefly describes each course, its audience and goals, how patterns were used in the class, and what I learned from the informal and formal feedback I collected.

Course: Introduction to HCI

This course was reported on in [1] before; the findings are given here just for reference.

During Using HCI patterns in a course this HCI design course for first-year computer science undergraduates in the Summer 1999 term at the University of Ulm in Germany, I spent one lecture of 90 minutes altogether dealing with HCI patterns: the idea of patterns, their origin in architecture, and their use for capturing HCI design concepts were explained, and copies of Jenifer Tidwell's *Common Ground* HCI pattern collection handed out. Students then took about 15 minutes to study the collection, and to find patterns that they could relate to their first own user interface prototyping exercise on which they were working at that time.

Informal Positive informal feedback feedback during this exercise, and in the following week while the prototypes were finished, was very encouraging. Most students were able to immediately relate several patterns to problems they had been facing during their design themselves.

This Formal survey was confirmed in a formal statistical evaluation, which was carried out two weeks after the above lecture, in an unannounced lecture evaluation. Students were asked to rate various aspects of the lecture, including the following questions concerning the design pattern approach presented:

1. I remember the following HCI design patterns:
2. For the overall understanding and remembering of user interface design concepts, the patterns were (1=very useful . . . 5=completely useless).
3. I was able to find problems and solutions for our own design project in the pattern collection (1=absolutely . . . 5=not at all).
4. I can imagine using this pattern concept in future design projects (1=certainly yes . . . 5=certainly not).

Results and Discussion

$n_0 = 32$ students filled out the questionnaire; of these, $n = 26$ answered the questions about patterns. The results are shown in Fig. 1.

On Students remember patterns average, $\mu \approx 1.73$ patterns were remembered, with a standard deviation of $\sigma \approx 1.65$. Lecturers will agree that this is quite promising, considering that students only spent relatively short time with the material during the lecture, only looking at a few patterns in any detail, and that the material had not been revisited by students for the final examinations yet. The vocabulary function of HCI design patterns seems to have succeeded quite well. The large standard deviation reflects the fact that several students wrote down no patterns at all, an effect that does not fit into the standard distribution; with an examination-like test situation, they may have spent more time trying to remember some of the patterns.

The Students consider HCI patterns useful usefulness of the pattern language for understanding HCI design issues was rated with an overall $\mu \approx 1.96$, i.e., with the second-best grade possible, with a relatively small standard deviation of $\sigma \approx 0.65$, indicating a high level of consensus among the students.

Usefulness for current project work was rated slightly worse,

Position paper for "Patterns in Practice: A Workshop for UI Designers", workshop at *CHI 2002 International Conference on Human Factors of Computing Systems (Minneapolis, MI, Apr 21-25, 2002).* To be published.

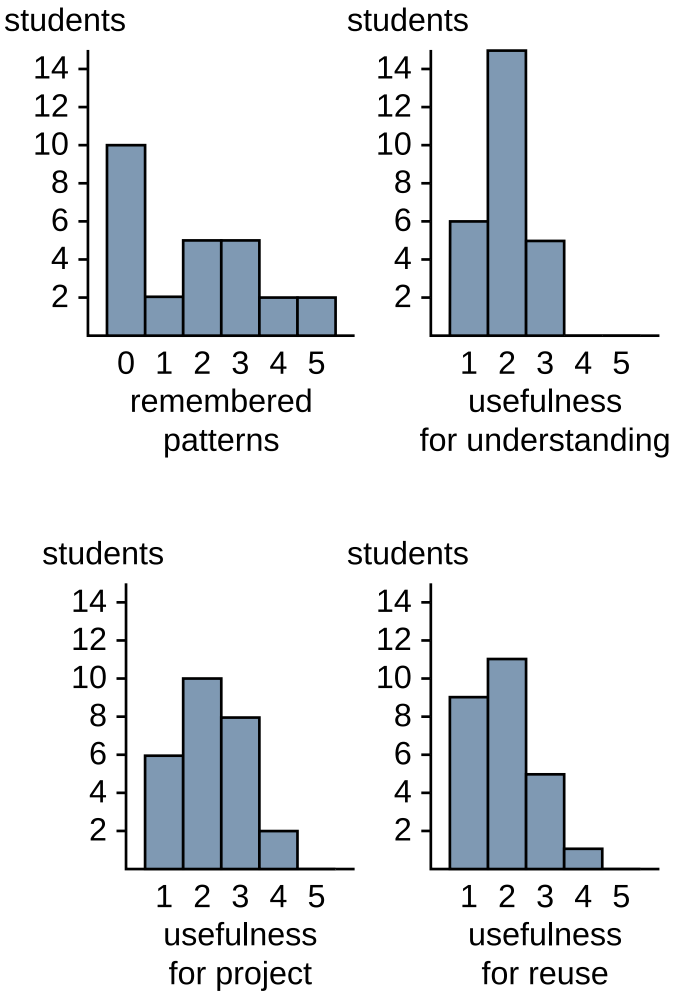

Figure 1: Results of the patterns survey, showing how many patterns were remembered, and their perceived usefulness for learning, current work, and future projects.

but still with an overall second best grade ($\mu \approx 2.23$). A slightly higher standard deviation ($\sigma \approx 0.89$) shows that there was less consensus on this question.

Finally, the confidence that this pattern concept would be reused in future projects was again quite high ($\mu \approx 1.94$), with relatively great consensus ($\sigma \approx 0.81$).

In all, these results indicated that a pattern approach in HCI education would be useful and convincing. Through the structured combination of widely known examples with generalized recommendations, even first-year undergraduates were able to quickly relate to this format, and found it useful and worth considering for their further projects.

Course: Patterns In Interaction Design

I held this course at Stanford University as part of the computer science curriculum in Spring 2001. As a quarterly course, it consisted of eight weeks of classes at 110 minutes each. The audience consisted of 18 students with a variety of backgrounds—8 undergraduate students majoring in fields ranging from (mostly) CS and Symbolic Systems, to Education and Arts, to undeclared freshmen; 5 CS Master's students; 3 MA students from the Learning, Design, and Technology program; a Psychology PhD student, and a postdoctoral visitor.

The course used [1] as its main textbook, and was structured around that book and additional readings, with the following topics covered per class:

Introduction. HCI, problems of interdisciplinary and participatory design, how and why to capture HCI design experience.

Off to Alexandria: Patterns in Architecture. Alexander's work, the STREET CAFE sample pattern.

I Strayed from the Path: Patterns in Software Engineering. Adopted and lost characteristics of the original pattern idea in software engineering, Gamma's ABSTRACT FACTORY sample pattern, writer's workshops.

Back to Basics: The Structure of Patterns. Defining the components of patterns and pattern languages.

Been There, Done That: Existing HCI Design Pattern Languages. Comparing existing collections such as those by Tidwell [3], Borchers [1] and van Welie [4].

Interdisciplinary Patterns. Using multiple pattern languages from stakeholders' disciplines to support communication in interdisciplinary design teams.

"So?" Using Patterns in the Development Process. Applying the pattern principle to user discovery, guideline formulation, design rationale, and other parts of the design and development process.

The Future of HCI Patterns: Outlook and Review. Recent research results, e.g., computer tools for authoring and disseminating HCI design patterns.

The course was accompanied by a series of assignments that asked the students to write their own patterns. Initially, before having introduced the pattern concept in full, those were *proto-patterns* that did not have a complete pattern structure yet, but that already aimed to capture recurring successful design solutions identified by the students in the world around them. Those were followed, according to the class topics, by architectural patterns and software patterns, before getting to actual HCI design patterns. That way, the students experienced the same historical development that the HCI Design Pattern concept itself went through.

After a group assignment of creating an HCI pattern language, the interdisciplinary assignment asked the students to choose a project they had recently worked on, and to explore whether they could identify patterns from the application domain, the HCI design, and the software engineering areas of that project. Several core patterns had to be formulated in full (reusing earlier patterns was possible), while their contextual patterns were only due as sketches of problem and solution statement.

Finally, students were asked to explore in essays some of the advanced topics, such as pattern editing toolkits, additional links and their semantics in pattern languages, success and failure stories of actual pattern use, or project plans that showed how patterns would be used in the various stages of a hypothetical follow-up project.

All pattern assignments were not only reviewed by the instructors, but also discussed during in-class *writers' workshops* in groups of four. This allowed the students to critique and learn from each others' pattern ideas, and allowed the groups for subsequent assignments to be chosen according to the individual patterns that the students had focused on.

Evaluation

Two measurements were available for this class. First, over the course of the class each student created about a dozen patterns that were subsequently workshopped, reviewed, and rewritten. The quality and characteristics of those patterns were a useful indicator to see how the patterns concept was received and understood. Second, students filled out a course evaluation questionnaire that is standard in the Stanford School of Engineering, which is an indicator of how the students judged the quality of the course. While this is only an indirect measure for how students judged the patterns concept, a course offering a topic that turns out to be of little perceived use will be judged accordingly.

Patterns

The patterns that students created were generally of fairly good quality. Initially, the course suggested to use the Alexandrian format. After the core characteristics of patterns had been explained in class in more detail, the format restriction was removed, merely asking for explanation why a different format had been chosen.

In all, the students had no problem understanding and applying the pattern format to their own contributions. Most student patterns had a good structure that contained the right kinds of content in the right place (examples can be seen at [2]). This seems to indicate that the clear, uniform, human-readable structure of patterns (as introduced in the textbook) can be understood and adopted by newcomers fairly easily.

One surprising result was that students immediately picked up the pattern writing style; many of them actually even adopted Alexander's slightly baroque tone. As an example, here is the problem statement from an architectural pattern (second assignment) called RAILROAD WAITING ROOM by a junior undergraduate student (admittedly, he was from the Arts & Humanities Program :)):

> The railroad waiting room provides an area for both boredom and excitement, thus resolving two fundamentally opposed emotions. Every railroad station has a waiting room, and though it is often overlooked as the least important aspect of the railroad structure itself, it is in actuality the most important because of the dynamic environment it creates.

Students had obvious problems, on the other hand, to find the right level of granularity and abstraction in their patterns. We observed that students new to patterns tend to write patterns that try to capture more general concepts than what patterns would usually address. Interestingly, in the architectural and software design patterns, this was less of a problem, while the HCI design patterns often were at a level that was close to generic guidelines, focusing on descriptive criteria for good user interfaces, but not providing much constructive help in the design process. Examples included patterns such as MATCH THE USER'S MODEL, or FEEDBACK. The latter had the following solution statement:

> Always provide feedback channels in a computer interface to aid the user in using the system. Feedback is not limited to what takes place on the screen. Visual, audio and haptic channels can all be used for feedback. Appropriate use of feedback reduces confusion and stress, and makes the user more confident in using the interface. A user who knows what is going on is a happy user.

Most patterns, however, addressed about the right level of abstraction for a good, timeless pattern, and there were hardly any HCI design patterns that seemed too concrete (although most patterns focused on current graphical user interfaces, as opposed to UIs in general, which reduced their timelessness).

Questionnaire

All students were asked to fill out the standard course evaluation questionnaire of Stanford's School of Engineering. It asks students for ratings from 1 (excellent) to 5 (poor) on 18 questions that address topics such as quality of course organization, explanation of concepts and principles, apparent knowledge of material, responsiveness to class difficulty, etc.

Results from the 13 questionnaires returned showed that, in general, the course was rated slightly below, but similarly to other CS courses. To illustrate the level of similarity: The biggest difference was found in the question of motivation of students—in this case 1 to 5 meant that the course inspired extremely strong (1), strong (2), adequate (3), or minimal (4) effort, or eliminates motivation (5). The course rated 2.69

(i.e., adequate to strong) on this question (which was also the lowest result among all 18 questions), with the CS average at 2.14 (i.e., strong to adequate). All other differences were less significant than this.

Looking more closely at the results, they reflect some general characteristics of a typical course from a fresh research field: it performed above average for its organization and instructor's enthusiasm, but below average mostly for motivation (see above), perceived course value, readings, and fairness. Obviously, more work needs to be done in order to communicate not only the idea of HCI design patterns, but also their usefulness in the design process. The readings for the course (the textbook, plus several articles on interdiscplinary cooperation and HCI design patterns, as well as other collections such as Tidwell's HCI pattern language) need to be developed into a better collection, and the well-known problem of objectively judging patterns (much like creative writing and design work in general) needs to be addressed to arrive at a better measure for student success in such a course (as far as possible—the course shared the latter below-average rating for its fairness with other courses that teach students HCI principles by looking at examples).

Finally, it turned out that students found the workload and pace for the course to be significantly below their averages in the CS department. This indicates that less time would have been necessary for the material covered, or alternatively more depth been possible within the time allotted.

Summary

The above findings suggest the following:

1. Using HCI design patterns to teach HCI design principles is a useful approach that leads to above-average retention of design principles, and to a quick adoption of the pattern vocabulary, even among first-year undergraduates, lifting the design discussion to a more abstract, timeless, and efficient level.

2. Students consider the pattern format useful to formulate their own design experience. With some support, it may become the natural way in which new HCI students reason about and discuss HCI design decisions.

3. At the current level of existing research and pattern languages, HCI design patterns may best be used in HCI classes in two ways: When teaching HCI design patterns as a method, they should be embedded as a segment of a larger advanced class in HCI design methodologies. However, it also appears feasible to simply use the pattern concept as a tool and format in order to teach basic HCI design principles.

The "methods" way (as applied in the later Stanford class) describes a research- and methods-oriented class, in which case it seems that HCI design patterns can be covered as one of several alternative methods to describe HCI design experience. In the other, possibly more revolutionary case (as applied to some degree in the earlier Ulm class), students are just confronted with the pattern format as "the natural format" and medium that the teacher uses to explain HCI design principles, guidelines, decisions, and tradeoffs. Naturally, that second case requires this knowledge to already have been expressed as an HCI design pattern language, which in part contradicts the findings (see Tom Erickson's work) that suggest domain-specific languages to be more valuable than generic HCI pattern collections.

In all, it appears that HCI design patterns have great potential for teaching, not just as a topic, but particularly as a tool. I hope that the above observations help to inform future instructors about potential and pitfalls of teaching HCI (with) patterns.

REFERENCES

[1] Jan Borchers: *A Pattern Approach to Interaction Design.* John Wiley & Sons, New York, 2001, http://www.hcipatterns.org/paid/.

[2] Jan Borchers: *Patterns in Human-Computer Interaction Design.* CS377C course details, Computer Science Department, Stanford University, Spring 2001, http://www.stanford.edu/∼borchers/cs377c/.

[3] Jenifer Tidwell: *Common Ground: A Pattern Language for Human-Computer Interface Design,* http://www.mit.edu/ ∼jtidwell/interaction_patterns.html

[4] Martijn van Welie: *Web Design Patterns.* http://www.welie.com/patterns/index.html