

Patterns, Anti-Patterns and Guidelines – Effective Aids to Teaching HCI Principles?

Paula Kotzé
School of Computing
University of South Africa
Pretoria, RSA
+27 12 429 6677
kotzep@unisa.ac.za

Karen Renaud
Department of Computing
Science
University of Glasgow
Glasgow, UK
+44 141 3302052
karen@dcs.gla.ac.uk

**Kostas Koukouletsos, Babak
Khazaei, Andy Dearden**
Communications and
Computing Research Centre
Sheffield Hallam University
Sheffield, UK
+44 114 225 4951
Konstantinos.Koukouletsos@
student.shu.ac.uk, {B.Khazaei,
A.M.Dearden}@shu.ac.uk

ABSTRACT

Patterns have been promoted as a knowledge transfer mechanism in many domains, including software engineering, business process management, and more recently in the field of interaction design. In software engineering a concerted effort is also being made to identify and document anti-patterns for capturing expert knowledge and transferring this to novices. This paper reports on a review of reported studies of the use of patterns and anti-patterns in teaching software engineering and human-computer interaction, and also reports on three studies examining the impact of using guidelines, patterns and anti-patterns in teaching interaction design principles.

Keywords

Human-computer interaction, patterns, anti-patterns, guidelines, teaching strategies

1. INTRODUCTION

This paper is not about the merits of using patterns in interactive system development – much has been written about this already. This paper takes a critical look at using patterns and anti-patterns in teaching and debates the value of, and possible pros and cons of using *various forms of patterns and guidelines in teaching human-computer interaction (HCI) concepts*. One of the challenges in HCI teaching is to develop effective techniques for communicating the knowledge and experience of HCI experts. This is currently expressed in numerous texts, but is rapidly being converted into pattern format. If anti-patterns follow the historic trend of patterns, we should expect to see an escalating interest in the use of anti-patterns in HCI, and, in particular, their use in *teaching* HCI. Whereas patterns are no longer seen as foreign in HCI practice and teaching [22], the emergence of the anti-pattern literature requires due consideration of pedagogical issues. When an expert is involved in any project s/he will often act as a brake on the ambitious but ill-advised efforts of inexperienced designers. Hence it appears to be sensible to capture the knowledge of ‘what not to do’ as well as the knowledge of how things ought to be done. However, before the anticipated arrival of HCI anti-patterns, we should consider whether this logical next step is as beneficial as it might appear. We report a survey of examples of teaching ‘about’ (teaching certain patterns themselves) and ‘with’ (using patterns as a means to teach a wider topic) patterns and anti-patterns in both software engineering (SE) and HCI, and the success or failure thereof. We also address the ‘guidelines’ versus ‘patterns’ issue, and report on three studies we conducted using different forms of guidance. We conclude by analysing our findings and making recommendations for research and application.

2. CLARIFYING KEY CONCEPTS

Dearden and Finlay [9] distinguish between design advice in the form of heuristics, standards, guidelines, style guides, claims and patterns. Since the definitions of the concepts are often not distinct in literature, empirical work on the topic of the paper requires a clear scoping and definition of the terms used in the paper.

2.1 Patterns and Anti-patterns

Patterns are an approach to abstracting and capturing, for reuse, knowledge about what made a system, or paradigm, successful. A *pattern* is a text that provides ‘a structured description of an invariant solution to a recurrent problem within a context’ [9]. In SE, the initial concept of a design pattern has given rise to the concept of an ‘anti-pattern’. An *anti-pattern* describes a frequently used solution to a problem that generates ineffective or decidedly negative consequences, i.e. it is a solution that looks like a good idea, but which backfires when applied in the wrong context and can cause a project to fail [3, 21]. The concept has also attracted attention in interaction design (ID), e.g., Griffiths et al. [14] discuss anti-patterns and Graham [13] includes examples that he refers to as anti-patterns. In both these cases the term *anti-pattern* refers to a specific piece of negative advice or an example of bad practice, without explicitly associating any details on how to correct (refactor) the problem. To clarify our work, we distinguish between two types of anti-patterns:

- A *simple anti-pattern* tells the reader how to go from a problem to a poor solution [6]. Simple anti-patterns therefore

focus on presenting negative solutions.

- An *amelioration anti-pattern* tells the reader how to go from a problem to a bad solution, but also how to get from that bad solution to a good solution. It defines a migration (or refactoring) from negative solutions to positive solutions. It tells you why the bad solution looks attractive, why it turns out to be bad in conjunction with the desired new outcome or behaviour, and what positive patterns are applicable instead [4].

In this paper our main focus is on the use of patterns in teaching HCI. However, we will also discuss patterns in SE and in pedagogy. To clarify this area we define the following categories of patterns (and by analogy, simple anti-patterns and amelioration anti-patterns):

- *General software design patterns* (SE domain): a problem is stated in terms of desirable qualities of the internal structure and behaviour of software, and the solution is stated in terms of suggested code structures [9].
- *Interface software design patterns* (SE and HCI domain): a problem is stated in the domain of desirable interaction behaviours, and the solution is stated in terms of suggested code structures [9].
- *Interaction design patterns* (HCI domain): a problem is stated in the domain of human interaction issues, and the solution is stated in terms of suggested perceivable interaction behaviour [9]. Several collections of ID patterns exist, e.g., the Tidwell's collection [25], Sally Fincher's Pattern Form Gallery [12] and Van Welie's collection [28], Sari Laakso's collection [19], Coram and Lee's Experiences [7], amongst others.

Finally, we note the existence of pedagogical patterns. However, these are outside the scope of this paper.

- *Pedagogical Patterns*: a problem is stated in the domain of pedagogy, and the solution is stated in terms of actions, resources or activities that an educator can apply in their practice [24].

At this stage, publications about general software design *anti-patterns* abounds [6] but none of the major HCI pattern collections explicitly contain simple ID anti-patterns or amelioration patterns. Although the basic idea of HCI patterns and anti-patterns can be viewed as similar to that of SE, there are some fundamental differences [26]. SE patterns are primarily used by software developers 'all speaking the same language', while HCI patterns are also used for communication across disciplines, e.g., in interdisciplinary teams with human factors experts, users, interface designers and software developers. Unlike SE patterns, our research presented little evidence that HCI patterns are being used to generate application code (i.e. ID patterns leading to interface software design patterns).

2.2 Patterns and Guidelines

A distinction can also be drawn between patterns and guidelines. Dearden and Finlay [9] review different features that authors discussing patterns in HCI and SE have highlighted, with some key common characteristics being:

- A pattern implies an artefact. E.g., Borchers [1] describes general guidelines as *descriptive*, stating desirable general features of a 'good' finished interactive system, and patterns as *constructive* in that they suggest a solution to a problem.
- A pattern includes its rationale. E.g., Griffiths and Pemberton [15] claim that patterns record the meta-information surrounding the simple imperative instruction generally found in guidelines. This involves setting the context, drawing attention to the problem that the pattern aim to solve, envisaging the solution by resolving interacting social and physical relationship conflicts, and finally allowing for degrees of confidence in the invariance of the pattern.
- A pattern is grounded in a domain. E.g., Borchers [1] presents three separate pattern languages in different domains, with none of the patterns being held in common between the languages.
- A pattern is part of a pattern language, in which individual patterns are linked to other patterns, with the intention of guiding the reader of the language through a series of related design decisions. E.g., Dearden et al. [10] demonstrate how such a language can be used to structure participatory design sessions with users.
- A pattern bridges many different levels of abstraction, including both descriptions and examples.

In contrast, a guideline does not include its rationale, may omit specific illustrative examples, and is generally independent of the other guidelines within any set of guidelines.

3. A SURVEY OF TEACHING PATTERNS AND TEACHING WITH PATTERNS

Studying patterns in HCI and HCI-pedagogy necessitates a cognisance of the more mature pattern development in SE to understand and anticipate pattern development and maturation in HCI. We surveyed literature for examples of using patterns in teaching SE and HCI concepts, and also the teaching of patterns itself. The survey delivered mixed results. Although we did find pedagogical patterns for the teaching of SE engineering concepts, we found little recorded evidence that patterns are actually used in teaching, and if used, any assessment of how successful they were as teaching medium.

The most comprehensive study reported in the field of SE consists of several related individual studies done in the early 1990's by a group at the University of California, led by Linn and Clancy [5]. This is also the only study we could find that actually evaluated the effectiveness of the teaching methods they used and/or compared it to other methods. They studied the use of 'programming plans' in teaching programming and programming languages to novices. The group's overall finding was that programming plans need rich connections to examples and multiple links to context of use if they were to be effective in teaching. Cybulski and Linden [8] used multimedia and patterns to improve the effectiveness of the

teaching of object-oriented systems analysis and design to large groups at the University of Melbourne. They found that patterns, examples and case studies can be used as vehicle to teach techniques such as the analysis of problems, identification of alternative solutions to these problems, and the selection of the 'best' solution from a set of alternatives. Dugan [11] reports on the teaching of software performance engineering to undergraduates and proposes a course outline that concludes with review of software performance principles and guidelines, and patterns and anti-patterns related to performance. Lewis et al. [20] report on their research into ways to improve students' success in mastering design concepts through the use of object-oriented programming and design patterns and examine potentially effective teaching methods to teach patterns to novice programmers, and on a survey amongst experts to decide which patterns to include and the order in which these resulting pattern groupings should be taught. Sterkin [23] promotes an integrated approach for the systematic teaching of object-oriented design patterns and sees the goal of teaching design patterns as teaching pattern thinking rather than a particular subset of patterns. Weiss [29] does not cover design patterns per se in his CS1 course, but rather tries to motivate the underlying principles by using running case studies, and reveals only after the fact the identity of the design pattern, without making a big deal of pattern names. The students build on an evolving example using patterns and solutions to problems.

Although the list above is not claimed to be complete or comprehensive, it illustrates current trends. None of these studies has explicitly examined the effectiveness of teaching with patterns, in terms of student learning. What is interesting is that although we found evidence that SE *patterns* are used in teaching, we found *no* evidence that *anti-patterns* are actually taught or used in teaching (apart from a topic in the Dugan course). Why is this so? Is there a problem with teaching SE anti-patterns or teaching with SE anti-patterns, has nobody tried it, or have the results all been negative?

We also researched examples of current practice in teaching HCI patterns or using HCI patterns in teaching. Griffiths and Pemberton [16] discuss three approaches to incorporating design patterns into teaching: teaching *about* patterns, teaching *with* or *through* patterns, and *discovering* patterns. In teaching *about* patterns they focus on making connections between software design and design theory, and for teaching *through* patterns they use of Tidwell's pattern language [25] setting students practical exercises in the use of a pattern language in the process of design. The *discovering* patterns option delivered an interesting result. The patterns approach was explained in a lecture, after which students were encouraged to discover interface patterns for themselves by modifying existing guidelines and drawing on personal experience. All the examples produced by students were derived from bad design examples, and the majority of these stayed at the level of un-generalized examples. The authors hypothesized that their students may not have had sufficient experience and skill in the kinds of abstraction and analogical thinking required for effective pattern generation. Borchers [1, 2] used ID patterns to teach first year computer science students giving them a pattern collection to refer to in their design projects. He comments on the speed at which his students discovered patterns that were applicable to their situation, and how the students used a vocabulary of patterns to discuss designs and solutions between them. He later taught ID patterns to a class of mixed-level students from various backgrounds. The students had little problem in understanding and applying the pattern format, although they also experienced problems with finding the right level of granularity and abstraction in their patterns and often captured more general concepts close to that of generic guidelines. His overall conclusion was that using HCI patterns to teach HCI concepts is a useful approach leading to above average retention of design principles (although he does not state that he actually measured this and against what); that students consider the pattern format useful to formulate their own design experience; that the teaching HCI design patterns should be embedded as a segment of a wider class in HCI design methodologies; and that it seems feasible to use the pattern concept as a tool to teach basic HCI design concepts. Seffah [22] advocates the use of patterns as both a useful tool (teaching with patterns) and as a learning resource (teaching patterns themselves) in teaching human-centred design skills to software to SE students. Although the paper mentions feedback from students, he also does not state whether he has assessed learning taking place.

Although these examples refer to teaching with ID patterns, again we found no evidence exploring the use of anti-patterns in teaching HCI concepts. Can this be attributed to the same reasons as those causing SE anti-patterns not to be taught?

3.1 Using SE Patterns in Teaching – Our Own Experience

Following the outcome of our survey we still didn't know whether they are being used with equal maturity and efficacy in teaching. To examine patterns in teaching, we conducted a study on the use of patterns in SE teaching with a group of third year SE students at the University of Glasgow. Students were randomly allocated to groups of 5 to do a project entailing design and implementation of a project management system. Students were taught some basic SE patterns and provided with examples of interface software design patterns. Even though it was a group project, each student had to write an individual report about what s/he learnt during the project. The report included the following: what they did on the group exercise; their main individual achievements on the exercise; individual problems encountered during the exercise and how they were handled; what was learned about software design (recommended topics included the role of patterns and refactoring). It was possible to use at least 5 patterns in the exercise. In assessing the reports, it was found of the 96 students in the class, 60 students made no mention of patterns; 6 students mentioned patterns but only in a theoretical way, merely reproducing the pattern definition; 3 students did not use patterns although they mentioned them in passing. Of the 27 students who reported using patterns, 14 students used only 1 pattern; 10 students used only 2 patterns; 2 students used 3 patterns; and only 1 student used all 5 patterns. Only 28% of the students made use of patterns in their group project. Analysing the reports we found it puzzling that the team members of the student who reported using all 5 patterns, did not report using the same 5 patterns. It is possible that students made use of patterns and then did

not report on it in their individual reports, but this is unlikely because it was an explicitly mentioned topic. The only conclusion we can draw from is that students had the theoretical knowledge but had difficulty applying it.

The outcome of this study, combined with the results of our survey, left us with even more questions than answers. Can we really teach effectively using patterns and will students be able to apply the concepts we are trying to convey? What will happen if we start teaching with anti-patterns (or negative guidelines)? If teaching with patterns are claimed or proven to be successful (or even unsuccessful), can we expect knowledge transfer to be the same for anti-patterns?

4. HCI POSITIVE GUIDELINES AND HCI NEGATIVE GUIDELINES

To try to answer the positive-negative question we conducted two experiments using positive and negative guidelines to represent patterns and anti-patterns, respectively. We briefly discuss the results of these experiments below.

4.1 First Study – Guidelines versus Anti-guidelines Intra-group Study

Our first study investigated the difference in the effectiveness of positive guidelines, negative guidelines and the concatenation of an anti-pattern followed by a pattern (this study is described in detail in Van Biljon et al. [26]). The case study involved a single group of 66 honours-level students taking a course in database and Internet technology. One of the lectures of the course concentrated on e-commerce, and presented various concepts both in positive and negative format. The students were required to form groups of 3 and to develop an e-auction site given the guidelines. The positive and negative guidelines were presented and assessed separately. First the whole class was given some positive guidelines relevant to the domain. Next the students were given some negative guidelines addressing different aspects of the design. Finally, the group was exposed to a third block of guidance presented as simple anti-patterns followed by a positive guideline. After each presentation the results (whether they applied the guidelines or not) were recorded. Intra-group comparison was used to analyse the students' performance relating to the use of the various guidelines or the anti-pattern. The results indicated that the negative guidelines had limited success in influencing behaviour, while the positive guidelines appear to be more effective for student learning, but only if the guideline is the students' first encounter with the design problem. In the case of the anti-pattern followed by the positive guideline the influence of the previously experienced anti-pattern was so strong that most of the students were unaffected by the positive guideline. There may, of course, be other factors which influenced the students. Being intra-group, the results raised some concerns and justified further investigation with another group, as reported below. However, the finding that positive guidelines are more effective in supporting learning than negative guidelines, is in line with theories of mental models that suggest a mental model represents explicitly what is true, but not what is false, leading naive reasoners to make systematic errors [17]. Although this case study was based on enriched guidelines, we can argue that it might show that anti-patterns based on the templates for patterns would not necessarily provide the same benefit because the cognitive processing of anti-patterns has to deal with negation.

4.2 Second Study – Guidelines versus Anti-guidelines Structured Inter-group Study

Based on the findings and limitations of the first study we planned a second more structured experiment. The class was split up into two distinct groups. We presented information (on a non-essential topic) to the two groups differently to support inter-group comparison. The experiment was designed to test the following hypothesis: *There is a significant difference between the performance of students taught positively using patterns and those taught negatively using anti-patterns.* The experiment is described in detail in Van Biljon et al. [27]. We used two pre-established tutorial groups and designated the one as the *pattern group* and the other as the *anti-pattern group*. The group members were third year Computing Science students at the University of Glasgow, with 27 students in each group, and the groups usually attended separate tutorial sessions. Group tutorials had information presented differently – positive guidelines or negative guidelines. We once again used sets of *guidelines* instead of full patterns in this experiment, but the guidelines were combined with examples making them closer to the textual form of patterns. Each group received instructions in a presentation about: the different colour wavelengths and characteristics of colour-blindness; error reporting and data entry instructions; a discussion of Fitts' Law and the implications for interface design. In presenting the material to the positive group, all information was presented in a positive format, e.g., *keep colours bright and bold*, and *put instructions next to the data entry field they apply to*. The negative group, on the other hand, received information presented in a negative form, e.g., *don't use pale or dark colours* and *don't put instructions where they won't be seen*. Students were also given examples of user instructions. The positive group were given examples with explicit instructions such as: *put instructions right next to the data entry field to which is applies*, and an example was shown: *Please enter a valid e-mail address in the format name@something.xx.xx*. The negative group was not explicitly told to provide instructions, but were shown examples of user interfaces which had instructions next to a data entry field while being instructed on what they should not do. Once the information was presented, students had to complete a sheet asking them to design a web page to allow students to register to receive marks by e-mail. They were asked to design the page and to label each item with the colour to be used and students in both groups were specifically instructed *not* to use black and white. The final question asked them how they would accommodate their knowledge of Fitts' Law into their interface. The answer sheets of the students were scored by awarding marks for correct use of taught design principles regarding the use of colour, the instructions given to the user, button size and button colour, and error reporting. Table 1 depicts the difference between the average scores of the students in the positive group and those in the negative group for each of the assessed components. The overall finding was that statistically the positive group performed significantly better than the the negative group.

	Use of Colour	Instructions given	Button Design	Error Reporting
Anti-Patterns	39	41	22	46
Patterns	47	54	37	59

Table 1: Comparing the marks of students in the anti-pattern group and the pattern group per component.

5. HCI GUIDELINES AND PATTERNS

In the first two studies, we represented patterns by means of sets of positive guidelines and associated examples, and anti-patterns by means of negative guidelines and associated examples. The rationale was that if there is a difference in the comprehension and processing of guidelines and anti-guidelines, this difference would be compounded in using patterns and anti-patterns. One of the criticisms levelled against our first two experiments was that we used enriched guidelines instead of full patterns, and that that learning might be improved by using full and comprehensive patterns. We identified this as a further research topic. To address this we searched the literature for studies comparing the use of guidelines with the use of patterns (unfortunately due to length limitations we cannot report on our findings here). We are also now studying differences between teaching with guidelines and teaching with patterns in a structured experiment, following on a pilot study. We briefly report on the outcomes of the pilot study.

5.1 Pilot study –Teaching Patterns vs. Teaching Guidelines

A selected set of patterns from the Van Welie collection [28] and a corresponding set of guidelines taken from Koyani et al. [18], were used to teach eleven second year undergraduate students, in an optional HCI module, usability principles and web interface design. Based on the test scores rankings in a preliminary test to evaluate their general usability sense and knowledge, we divided the students into two groups to create matched groups. One group was trained about usability principles and web interface design using guidelines, and the other group using the patterns. A booklet was distributed to the students in each group containing a set of patterns or a set of guidelines. Two one-hour tutorials were followed by a one hour post experiment test that included an evaluation and a design task. The evaluation task consisted of identifying flaws and usability problems and suggesting useful redesign ideas and improvements (at least 10) in two small sections of a web site. The design task consisted of designing and producing a paper prototype of two interlinked web pages with specific requirements regarding the content and the functionality of each page. After the test, each group received a one hour tutorial on the other technique to ensure all students were able to complete the module on an equal basis. In a final session patterns and guidelines were reviewed together and students reported their feelings, understanding and opinion about patterns and guidelines and their usefulness in the design process. Although the small number of participating students means that we are unable to draw any reliable conclusions comparing guidelines and patterns, our interim observations were consistent with the findings of Wesson and Cowley [30]. We found the guidelines easier to teach than patterns and also easier for students to comprehend and to remember – patterns are longer in format than guidelines, and must be analysed in more detail before use; patterns seem to require a more careful and thoughtful teaching approaches if we want to realize their full potential; links between patterns need further attention in order to be appreciated by the students; the names of patterns and guidelines carry a significant weight (patterns like *Breadcrumbs* to assist navigation and guidelines like *avoid jargon* are easily remembered and easily applied, but guidelines with indistinguishable names like *provide users with good ways to reduce options* or *order elements to maximize user performance* do not convey any information and may misguide the students). The examples presented with each pattern, or guideline, are probably the part that captures the attention and interest of students. This was indicated by the majority of the students during the last teaching session. These examples help a user comprehend better the context and the intention of the pattern or guideline and provide an easy guide for the application.

6. CONCLUSION

We found that there is a great deal of hype but very little hard evidence about the benefits / pitfalls of using patterns in HCI education, and none on using anti-patterns in teaching. It is apparent from the findings of our studies that negative advice doesn't help students much. Students find negative advice in the form of enhanced guidelines more confusing than when presented with positive advice in the same format. Therefore negative examples (such as 'web sites that suck') should not form a major part of our teaching. If we generalize this to patterns compared with simple anti-patterns, we might deduce that students will find simple anti-patterns confusing, unless they are provided with clear alternatives (as is the aim of amelioration patterns). With regards to the methodology followed by our studies, we soon discovered that there is a real need to work with precise and clear definitions when investigating the use of patterns in teaching. The form and detail of current HCI-related patterns differ considerably, and what one author see as a pattern, is seen as a guideline with associated examples by another, etc. The same applies to terms such as guidelines, heuristics, style guides, or programming plans. There is therefore a desperate need to conduct solid studies of a suitable scale on the use of patterns and anti-patterns in teaching HCI concepts.

ACKNOWLEDGMENTS

We wish to thank Judy van Biljon of the University of South Africa for her contribution to our research. Much of the research reported in this paper was conducted while Karen Renaud was a guest lecturer at the University of South Africa, and while Kostas Koukouletsos was on leave from the Technological and Education Institute of Piraeus, Greece.

REFERENCES

1. Borchers, J.A. A pattern approach to Interaction Design. *AI & Society*, 15, (2001), p. 359 - 376.
2. Borchers, J.A., *Teaching HCI Design Patterns: Experience From Two University Courses*, in *Patterns in Practice: A Workshop for UI Designers (at CHI 2002 International Conference on Human Factors of Computing Systems)*. 2002: Minneapolis, MI.
3. Brown, W.J., Malveau, R.C., McCormick III, H.W., and Mowbray, T.J. *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. (1998), New York: John Wiley & Sons, Inc. 309.
4. Brown, W.J., Malveau, R.C., McCormick III, H.W., and Mowbray, T.J. *AntiPatterns*. 2002 [cited 2005-12-12]; Available from: <http://www.antipatterns.com/>.
5. Clancy, M.J. and Linn, M.C. Patterns and pedagogy. *ACM SIGCSE Bulletin*, 3/99, (1999), p. 37 - 42.
6. Cockburn, A., Baruz, A., Engelund, A., Hanes, P.B., Brown, C., Siska, C., Olson, D., Xexeo, G., Lowe, I., Chapman, J., Coplien, J.O., Holloway, J., Brown, K., Eichin, M., Phillips, R., Jeffries, R., Gordon, S., and McCormick III, H.W. *Antipattern*. 2005 2005-10-20 [cited 2005 2005-12-12]; Available from: <http://c2.com/cgi/wiki?AntiPattern>.
7. Coram, T. and Lee, J. *Experiences -- A Pattern Language for User Interface Design*. 1996 [cited 2005-12-20]; Available from: <http://www.maplefish.com/todd/papers/Experiences.html>.
8. Cybulski, J.L. and Linden, T., Teaching systems analysis and design using multimedia and patterns, In *Proceedings of the 13th Conference on Software Engineering Education & Training*. (2000).
9. Dearden, A. and Finlay, J. Pattern Languages in HCI: A critical review. *Human-Computer Interaction*, in press, (2006).
10. Dearden, A., Finlay, J., Allgar, E., and McManus, B., Using pattern languages in participatory design, In *Proceedings of PDC 2002*, T. Binder, J. Gregory, and I. Wagner, Editors. CPSR: Palo Alto CA. (2002).
11. Dugan, R.F., Performance lies my professor told me: the case for teaching software performance engineering to undergraduates, In *Proceedings of the 4th International Workshop on Software Engineering and Performance*. ACM Press: Redwood Shores, CA. (2004), p. 37 -48.
12. Fincher, S. *The Pattern Gallery*. 2000 2005-07-2005 [cited 2005-12-12]; Available from: <http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html>.
13. Graham, I. *A Pattern Language for Web Usability*. (2003), London: Addison Wesley.
14. Griffiths, R., Pemberton, L., Borchers, J., and Stork, A., Pattern languages for interaction design: Building momentum, In *CHI 2000 Extended Abstracts*. ACM Press. (2000), p. 363.
15. Griffiths, R.N. and Pemberton, L. *Don't write guidelines write patterns!* no date, [cited 2005-12-20]; Available from: <http://www.it.bton.ac.uk/staff/lp22/guidelinesdraft.html>.
16. Griffiths, R.N. and Pemberton, L. *Teaching Usability Design Through Pattern Language*. no date, [cited 2005-12-20]; Available from: <http://www.it.bton.ac.uk/staff/lp22/CHIpaper.html>.
17. Johnson-Laird, P. and Byrne, R. *Mental Models Website*. 2000 [cited 2006-01-15]; Available from: http://www.tcd.ie/Psychology/Ruth_Byrne/mental_models/theory.html.
18. Koyani, S.J., Bailey, R.W., and Nall, J.R. *Research-based Web Design and Usability Guidelines*. (2003).
19. Laakso, S.A. *User Interface Design Patterns*. 2003 2003-09-16 [cited 2005-12-21]; Available from: <http://www.cs.helsinki.fi/u/salaakso/patterns/>.
20. Lewis, T.L., Rosson, M.B., and Pérez-Quiñones, M., What do experts say? Teaching introductory design from an expert's perspective, In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*. ACM Press. (2004), p. 296 - 300.
21. Long, J. Software reuse antipatterns. *Software Engineering Notes*, 26, 4, (2001), p. 68 - 76.
22. Seffah, A. Learning the ropes: human-centered design skills and patterns for software engineers' education. *interactions*, 10, 5, (2003), p. 36 -45.
23. Sterkin, A. *Teaching design patterns*. no date [cited 2005-12-23]; Available from: http://www.hadassah-col.ac.il/cs_site/staff/asterkin/advCPlusProg/Teaching%20Design%20Patterns.pdf.
24. The Pedagogical Patterns Project. *Pedagogical Patterns*. no date [cited 2005 2005-12-12]; Available from: <http://www.pedagogicalpatterns.org/>.
25. Tidwell, J. *UI Patterns and techniques*. 2004 [cited 2005-12-19]; Available from: <http://time-tripper.com/uipatterns/>.
26. Van Biljon, J., Kotzé, P., Renaud, K., McGee, M., and Seffah, A., The use of anti-patterns in human computer interaction: wise or ill-advised? In *Fulfilling the promise of ICT*, G. Marsden, P. Kotzé, and A. Adesina-Ojo, Editors. SAICSIT (ACM Conference Proceedings Series): Pretoria. (2004), p. 176-185.
27. Van Biljon, J., Renaud, K., and Kotzé, P. Don't Do This ... Pitfalls in using Anti-Patterns in Teaching Human-Computer Interaction Principles. *Computers & Education*, (under review).
28. Van Welie, M. *Patterns in Interaction Design*. 2005 [cited 2005-12-19]; Available from: <http://www.welie.com/>.
29. Weiss, S., Teaching design patterns by stealth, In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. ACM Press. (2005), p. 492 - 494.
30. Wesson, J. and Cowley, N.L.O. *Designing with patterns: Possibilities and pitfalls*. 2nd Workshop on Software and Usability Cross-Pollination: The Role of Usability Patterns 2003 [cited 2005-12-23]; Available from: <http://wwwswt.informatik.uni-rostock.de/deutsch/Interact/05WessonCowley.pdf>.