Position paper for CHI 2000 Design Patterns Workshop

Sari A. Laakso, Karri-Pekka Laakso, Asko Saura

Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, FINLAND
+358 9 191 44268
{kpalaaks,asaura}@cs.helsinki.fi

Background

The idea of design patterns was originally presented by Alexander et al. [1,2] who collected frequently occurring town planning and architectural problems and presented concrete reusable solutions. They extracted good solutions from existing designs, but the domain of user interface design is currently even more challenging; contrary to architecture, we do not have a tradition of hundreds of years. At the present, user interface design is still in its infancy, and many solutions to very elementary problems have been strongly restricted by technical limitations.

During the last four years, we have taught user interface design to approximately 650 university students, and guided and evaluated altogether 350 one-term projects covering 40-50 different topics, e.g. News on demand, Intelligent home, Classroom reservations and Curriculum planning. Each project group has arranged a pilot usability test, and all the 350 test sessions have been analyzed and evaluated by us. During the process, we have found out several frequently recurring user interface design problems and sets of usable and non-usable solutions, i.e. user interface design patterns and anti-patterns.

This work has led us to develop a collection of user interface design patterns to be used as a tool and learning aid for user interface designers. To achieve this, we have extracted common problems and different solutions both from our students' projects and from existing software, and simultaneously designed new solutions. We have adapted some patterns from existing pattern repositories, Jenifer Tidwell's Common Ground [5] in particular, for our own needs. Our collection currently includes 25 patterns and pattern candidates. Some of the patterns are closely interlinked, others are independent design tools.

Benefits of patterns

We believe that the advantages of employing user interface design patterns are similar to those using software design patterns [4]. They provide a common vocabulary to communicate, document and explore design alternatives. During design or documentation, you do not have to describe the design patterns you use, because the descriptions can be found in the pattern collections, and anyone who is not yet familiar with a pattern can look it up there. You are able to talk about the design at a higher level of abstraction which results in a more efficient design process, and gradually you may raise the level of your design. Instead of just talking about buttons, menus and drop-down lists, you can talk about larger patterns and have more expressive tools for communication. Furthermore, learning patterns helps designers learn user interface design much faster, because the patterns provide concrete examples to common, recurring problems and they are easy to apply.

Our view of the concept 'pattern'

Software design patterns are thought to be problem-solution pairs with the emphasis on the solution, while Alexander et al. stress the problem descriptions of their architectural patterns. We view user interface design patterns as a set of problems for which some good solutions are known. We know there are problems yet to be solved, and have come to describe those problems as something like pattern seeds, too. Furthermore, existing software has so many anti-pattern solutions for the same problems, that it is useful to think of the problem as the link between different design solutions.

Experiences of applying patterns

We have already used our present collection of design patterns as a set of tools in our advanced course for user interface design. The qualitative results were remarkable. In some extreme cases, the students were able to create a better user interface design for a particular design problem in four hours than less experienced students were able to produce during four months. The result was provided by expert evaluation, not usability testing.

Right from the start, we have applied the patterns in real software projects. The benefits of efficient designer-to-designer communication immediately became apparent, and initial designs have consistently improved. We do not yet have sufficient experience to say how much of an overall effect the patterns have on actual projects.

Conclusions

We have yet to determine a working method to extract and write down the patterns, and many of them only exist quite informally. As a result, we are very interested in learning how to document user interface design patterns in a meaningful way.

References

1. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.A. Pattern Language. Oxford University Press, New York, 1977.
2. Alexander, C. The Timeless Way of Building. Oxford University Press, New York, 1979.
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, Massachusetts, 1994.
4. Tidwell, J. Common Ground: A Pattern Language for Human-Computer Interface Design. http://www.mit.edu/~jtidwell/interaction_patterns.html