ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/238591781

PEICS: HCI patterns for the design of interactive systems

Article · June 2011

DOI: 10.1145/2018431.2018432

CITATIONS
0

READS
59

6 authors, including:

Gerrit Meixner
Hochschule Heilbronn
99 PUBLICATIONS **385** CITATIONS

SEE PROFILE

Ahmed Seffah
Lappeenranta – Lahti University of Technology LUT
204 PUBLICATIONS **2,845** CITATIONS

SEE PROFILE

Kai Breiner
Fraunhofer Institute for Experimental Software Engineering IESE
26 PUBLICATIONS **112** CITATIONS

SEE PROFILE

Peter Forbrig
University of Rostock
228 PUBLICATIONS **968** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Cooperative driver vehicle interaction (KoFFI) View project

EVElyn View project

All content following this page was uploaded by Gerrit Meixner on 14 January 2016.

The user has requested enhancement of the downloaded file.

PEICS: HCI Patterns for the Design of Interactive Systems

Marc Seissler
University of Kaiserslautern
Center for Human-Machine-Interaction
67663, Kaiserslautern, Germany
Marc.Seissler@mv.uni-kl.de

Kai Breiner
University of Kaiserslautern
Software Engineering Research Group
67663 Kaiserslautern, Germany
Breiner@cs.uni-kl.de

Gerrit Meixner
German Research Center for
Artificial Intelligence (DFKI)
67663 Kaiserslautern, Germany
Gerrit.Meixner@dfki.de

Peter Forbrig
University of Rostock
Department of Computer Science
18051, Rostock, Germany
Peter.Forbrig@uni-rostock.de

Ahmed Seffah
University of Troyes,
Institut Charles Delaunay,
10010 Troyes Cedex, France
Ahmed.Seffah@utt.fr

Kerstin Klöckner
Fraunhofer Institute for Experimental
Software Engineering (IESE)
67663 Kaiserslautern, Germany
Kerstin.Kloeckner@iese.fraunhofer.de

ABSTRACT
In this paper, we summarize the results of the *Pattern Driven Engineering of Interactive Computing Systems* (PEICS) which took place at the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS) 2011.

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design Tools and Techniques – *evolutionary prototyping, user interfaces*. D.2.10 [**Software Engineering**]: Design – representation. D.2.11 [**Software Engineering**]: Software Architectures – *Patterns (e.g., client/server, pipeline, blackboard)*. D.3.3 [**Programming Language**]: Language Constructs and Features – *Patterns*. H.5.2 **[Information Interfaces and Presentation (e.g., HCI)]**: User interfaces –*Standardization, Ergonomics, Theory and methods*.

General Terms
Documentation, Design, Experimentation, Human Factors, Standardization, Languages.

Keywords
Descriptive pattern, Generative pattern, Interaction Design pattern, HCI pattern, Model-Driven Engineering, Model-Based User Interface Development, MBUID, Usability, Ergonomics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
PEICS'11, June 13, 2011, Pisa, Italy
Copyright 2011 ACM 978-1-4503-0856-4…$10.00.

1. INTRODUCTION
Ensuring ergonomic quality of user interfaces (UI) is one challenge that has to be addressed by todays' UI designers and developers. HCI pattern languages are databases of design knowledge presentation that have been introduced to support inexperienced developers via promoting proven solutions for recurring design problems.

One strength of patterns is that they describe the rational in which context (when) their solution should be applied, *why* the solution is used in this context and *how* the solution should look like.

But despite intense research activities in the last years, a *lingua franca*, a common language for the standardized description and organization of HCI patterns is missing. This hinders the design of suitable tools that support developers in identifying patterns and applying them accordingly to the given problem context.

To support the constructive and efficient use of HCI patterns in today's (model-based) development process the informal textual, or graphical notation of HCI patterns has to be overcome.

2. CHALLENGES
In the following we consider three major deficiencies of the current state of the art.

2.1 Formalization and Organization
One major shortcoming of today's pattern languages is the lack of formalization. Most of today's pattern forms like the *Pattern Language Markup Language* (PLML) [9], XPLML [13] and PLMLx [3] are based on informal problem and solution descriptions. This hinders the integration of such languages in model-based development processes. During the PEICS 2010 workshop [5] concepts have been presented that address this issue

via integrating user interface models for machine-readable pattern descriptions [6].

The degree of formalization also has an impact on the organization of patterns. Jordan Janairo [10] and Christian Kruschitz [12] proposed concepts how formalized HCI pattern relations and ontologies can be used to structure pattern languages. Besides interrelating HCI patterns, concepts have been proposed to formalize the solution part of HCI pattern and design pattern. Graph-based approaches, like introduced in [4], can be used to relate these patterns in a model-based development process. Appropriate pattern language structures are especially useful for novice developers, to support efficient identification suitable pattern solutions.

2.2 Tool-Support

Tool support is inevitable for allowing an efficient use of patterns. Although many tools for managing and applying patterns [7] do exist, there are only a few tools that support developers throughout the whole pattern application process, such as the "task pattern wizard" [15]. Due to informal pattern notation those tools still do not employ their full power, e.g. by supporting computer-supported reasoning for the most suitable design solution for a given problem context. Further, most of the tools are not integrated in the development processes and do not support the developers in the instantiation, adaption and integration of HCI patterns. This hinders the use of HCI patterns within semi-automatic processes during run-time generation of user interfaces [14].

2.3 Empirical Evaluation

Lastly, a pattern's benefit is related to the fact that pattern descriptions often lack empirical evidence that the pattern solves the problem at hand, e.g. to which extend it improves the usability of a user interface, or does a particular pattern affect the UX of another pattern. Typically, pattern descriptions just give examples of good instantiations of the pattern "in real life" in form of a screenshot or other media types.

Single point of failure is the description including a "confidence" section in which the pattern author declares the confidence level with which he "beliefs" that the pattern presents an invariant solution. However, this typically lacks what quality aspect is exactly improved by the pattern.

3. RESULTS & FUTURE DIRECTIONS

The PEICS workshop started with a short summary of the last year's workshop results and introduction round. After that, three presentations introduced different perspectives on how to use HCI patterns for engineering interactive computing systems.

Jan van den Bergh started the presentation session by introducing a concept that uses the CAP3 [2] approach to formulate patterns via a graphical diagram notation. Key of this approach is to capture the core of a pattern via abstracting from details, which are often found in today's pattern language examples like in [17].

Christian Märtin presented a concept for using User Experience (UX) patterns in the model-based development process PaMGIS [8]. Goal of this approach is to efficiently reuse the identified patterns when designing new systems in the same application domain.

Holger Röder was closing the presentation session by presenting a concept how to integrate usability aspects in use cases in the domain of requirements engineering. So far, 17 functional features have been identified that support developers to apply usability when eliciting system requirements.

One of the first discussion points addressed during the subsequent plenary discussion was how to communicate HCI patterns to developers. Like everything else in life, developers have to learn the foundations (and need to be convinced of the economic advantage) of HCI patterns before they can be used. To allow for an easy use of pattern languages, we have to consider that developers first should be trained how to read their structure and solution descriptions. Teaching how to use HCI patterns should therefore be part of today's lectures at universities. Within this context, most workshop participants identified the problem that when students are told to use HCI patterns, they often try to apply as much patterns as possible, instead of identifying where their application makes sense – which is hard to judge for inexperienced developers. This might imply that HCI patterns need a better concept which quality attributes they do address. This can help to prioritize patterns and select the most suitable pattern for a given problem. All workshop participants agreed that it is better to only apply a limited set of core patterns than as many patterns as possible.

This also might result from the fact that today the terms pattern, template and building block are not strictly distinguished. With respect to the original pattern definition by Alexander [1] the workshop participants agreed, that patterns should represent generic, core solutions on different UI abstraction levels and development phases. This implies that patterns are a restricted set of heterogeneous, proven design solutions which can be refined via pattern instances e.g. in form of templates. These templates can represent a variant of the pattern but implement the core idea of the underlying pattern.

Another topic of the workshop was that in industry most companies claim that usability is very important, but they do not align their processes to this goal. This problem also fits on the use of HCI patterns. To benefit from HCI patterns their integration in the development process is crucial. This means that they have to be easy accessible for the developers and – maybe most important – up-to-date. While tools can help to address the first point new evaluation methods are needed to prove the usability of patterns.

Most pattern languages/libraries haven't been proved if their solutions are suitable to address the identified problems. Hence, we need a set of quality criteria (e.g. GQM [16]) that can be used to prove the pattern's benefit. These criteria can be used in order to assess the pattern's effect during evaluation, as well as enlisting them in the pattern description to enhance the "credibility" of the solutions. Such criteria could also provide new search terms which can be used to find a pattern with an appropriate solution. A first step in this direction has been made by defining a quality model for user experience (patterns) for business applications [11].

Other interesting discussion points, which will be addressed in future work, were how to define, capture and illustrate User Experience (UX) patterns and how to integrate patterns in agile development processes.

4. ACKNOWLEDGMENTS

We thank all workshop participants for their cooperation, ideas and fruitful discussions during the workshop.

5. REFERENCES

[1] Alexander, C., Ishikawa, S. and Silverstein, M., 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press,.

[2] Van den Bergh, J., Luyten, K. and Coninx, K. CAP3: context-sensitive abstract user interface specification. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, ACM (2011), 31-40.

[3] Bienhaus, D. PLMLx Doc. 2004. http://www.cs.kent.ac.uk/people/staff/saf/patterns/diethelm/plmlx_doc/index.html [Accessed on 20.12.2009].

[4] Bottoni, P., Guerra, E. and de Lara, J., 2010. Formalising design and interaction patterns and their relationships. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, ACM , 32-35.

[5] Breiner, K., Seissler, M., Meixner, G., Forbrig, P., Seffah, A. and Klöckner, K., 2010. Pattern-driven engineering of interactive computing systems (PEICS). *EICS*, 367-368.

[6] Breiner, K., Seissler, M., Meixner, G., Forbrig, P., Seffah, A. and Klöckner, K., 2010. PEICS: towards HCI patterns into engineering of interactive systems. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, ACM, 1-3.

[7] Deng, J., Kemp, E. and Todd, E.G., 2005. Managing UI pattern collections. In *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural*, ACM, 31-38.

[8] Engel, J. and Märtin, C., 2009. PaMGIS: A Framework for Pattern-Based Modeling and Generation of Interactive Systems. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends*, Springer-Verlag, 826–835.

[9] Fincher, S., Finlay, J., Greene, S. et al. 2003. Perspectives on HCI patterns: concepts and tools. In *CHI '03 extended abstracts on Human factors in computing systems*, ACM, 1044-1045.

[10] Janeiro, J., Barbosa, S.D.J., Springer, T. and Schill, A., 2010. Semantically relating user interface design patterns. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems - PEICS '10*, ACM, 40-43.

[11] Klöckner, K., Kohler, K., Kerkow, D., Niebuhr, S. and Nass, C., 2010. Aligning business goals and user goals by engineering hedonic quality. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, ACM, 241-250.

[12] Kruschitz, C. and Hitz, M., 2010. Bringing formalism and unification to human-computer interaction design patterns. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, ACM, 20-23.

[13] Kruschitz, C., 2009. XPLML: A HCI Pattern Formalizing and Unifying Approach. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, ACM, 4117-4122.

[14] Seissler, M., Meixner, G. and Breiner, K., 2010. Using HCI Patterns within the Model-Based Development of Run-Time Adaptive User Interfaces. In *Proceedings of the 11th IFAC/IFIPS/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems.*.

[15] Sinnig, D. et al., 2005. Patterns in Model-Based Engineering, In *Proc. of the 5th International Conference on Computer-Aided Design of User Interfaces*, 197-210.

[16] Solingen van, R. and Berghout, 1999. E. *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Improvement*. McGraw-Hill, Cambridge, UK.

[17] Tidwell, J., 2005. *Designing Interfaces*. O'Reilly Media.

View publication stats