USER INTERFACE PATTERNS: A FIELD STUDY EVALUATION

Regina Bernhaupt, Marco Winkler and Florence Pontico

Institute of Research in Informatics of Toulouse (IRIT), Groupe Interacting Humans with Computing Systems (IHCS)
118, Route de Narbonne, 31062 Toulouse, France
{bernhaupt, winckler, pontico}@irit.fr

ABSTRACT

The identification of user interface patterns, organization of patterns languages and the validation of pattern languages for interaction design have been an active research field in the area of human-computer interaction. However, only few researchers have explored the impact of pattern languages when used in the development process of interactive systems in real application domains. This work reports a field study on the development and evaluation of a pattern language in a company developing e-government applications. Conducting expert interviews combined with a field usability study we investigated the usage and adoption of a pattern language by the development team. Results show that pattern languages should take into account the idiosyncrasies of the application domain and results suggest that pattern languages can be a great leverage to improve usability culture in the industry.

KEYWORDS

Interaction design, pattern language, field usability study.

1. INTRODUCTION

Design patterns were first introduced in the field of architecture as a way of describing design solutions in an understandable and generalized form (Alexander et al., 1977). In recent years they have received considerable attention in the field of software engineering and in human-computer interaction (HCI) (Borchers, 2001). Attention was raised after many successful stories reported by the software engineering community on the development of design pattern for dealing with complex programming problems (Gamma et al, 1994). The interest in design pattern can be explained by their potential of recording and communicating design knowledge and supporting reuse of knowledge within the design process. Contrary to other strategies describing design knowledge (such as guidelines and heuristics) design patterns focus on solutions to specific problems rather than abstract suggestions. A design pattern is defined as ‘an invariant solution to a recurrent problem within a context’. They can be described single standing but they often appear as a set of interlinked patterns for a specific domain thus featuring a pattern language (van Welie and van der Veer, 2003).

Although many patterns and pattern languages have been published by the software engineering community and the HCI community, little has been reported about the actual usage of patterns in the industry (Dearden and Finlay, 2006). In order to understand the practical implications of design patterns in organizations we present a qualitative study based on an expert interview combined with a kind of field usability study (asking participants to solve some basic tasks to apply the pattern language). The case study was conducted in the industry, at SmalS-MvM, where we followed the implementation of user interface design patterns as a way to create a usability culture in that company. SmalS-MvM is devoted to the design, deployment, and handling of public e-Government applications. Our study is focused on the e-Government domain but some of the lessons learned could be generalized to the organization of patterns in general. This study discusses the actual use of design patterns after eight months after their implementation in an enterprise.

1 http://www.smals.be/
2. EVALUATING PATTERN LANGUAGES IN HCI

The HCI community has been prolific in the development of pattern languages for dealing with the idiosyncrasies of interaction design. Initial efforts exploring design patterns focused on the identification of patterns for different contexts. When the number of patterns collected increased, the debates included questions related to the organization of pattern languages in a meaningful way (Pontico et al. 2007). Currently, there is a wide range of pattern languages covering several areas in HCI, including web-based systems (van Duyne et al., 2002), safety critical systems (Grill and Blauhut, 2008), ubiquitous computing (Chung et al., 2004; Landay and Borriello, 2003), hypermedia applications (Rossi et al., 1999), and so on.

Literature on HCI pattern languages is abundant but only few works clearly states rules for ensuring a certain level of quality for pattern languages. Tood et al. (2004) mention three types of validation: (1) the validation of individual patterns, (2) internal validation (i.e. the connectivity between patterns in a collection featuring a pattern language), and (3) external validation (i.e. the impact of pattern languages in human activity and behavior). The validity of individual patterns is still an open issue as many authors report new patterns without discussing their applicability. However, a few exceptions exist. A good example of validation of emerging patterns is the Yahoo! approach (Malone et al, 2005) which defines that a solution should be successful used at least in two software systems before it is defined as a pattern. Kohler & Kerwow (2008) have investigated the usefulness of a pattern language devoted to workflow-based applications. They have investigated properties such as the understandability of individual patterns, if the patterns proposed in the catalogue are applicable to the target domain, if the provided solution really solves a problem, and if the provided solution can be effectively implemented. Unfortunately that study is limited to three experts that have never seen the pattern language before the interview and so the pattern language was evaluated outside real projects. Chung et al. (2004) have performed two-rounds of evaluations with 16 professional designers working in pairs. After each round of evaluation, patterns’ content and organization was improved according to the evaluation findings. These initial yet valuable efforts raise questions on how much validation (in terms of recurrence of problems identified, representativeness of participants, number of iterations in evaluations, etc) is required to validate a pattern.

Tood et al. (2004) proposed a method to analyze the internal validation of pattern languages. They consider the connectivity between patterns as a living document. The patterns are supposed to evolve over time and new patterns emerge according to the usage of the pattern language in the application context. However, none of the four pattern languages analyzed (i.e. van Welie’s GUI, Web and e-Commerce pattern languages, and Borchers’ HCI pattern language (2001)) was considered internally valid. According to the authors (Todd et al, 2004) three of the pattern languages should not be called pattern language because the pattern collections represent a very specific type of interface rather than a generic user interfaces.

Some empirical studies have been published concerning external validation criteria for the usage of user interface patterns in general (Cowley and Wesson, 2006). Several authors have claimed the use of patterns as a kind of lingua franca enabling multidisciplinary design teams to communicate and discuss their ideas for a particular design project (Erickson, 2000). The main argument behind this claim is that patterns are based on concrete examples and specific solutions so that they can be easily understood by anyone involved in design activities. This hypothesis has been explored by Finlay et al. (2002) in the context of participatory design to help users discuss the design possibilities with the designers. Other researchers have investigated design patterns as a mean to convey usability principles in teaching activities (Kotze et al, 2008). Another recurrent claim in the literature concerns the generative power of design patterns, which means they would help designers to create new solutions based on examples of actual design. In order to guide designers to the correct implementation of the proposed solutions, some authors have proposed to add diagrams (Bueno and Barbosa, 2006) and formal description techniques (Pontico et al. 2007) as part of the description of patterns. Lin and Landay (2008) have proposed a tool to exploit design patterns for sketching the user interface. Gaffar et al. (2005) have pushed the generative aspect further, having tool support for implementing design patterns as programmable artifacts using a XML-based language (2005). However, we could not find any report in the literature demonstrating that claims are valid in real projects. When empirical validation of these claims exists, they are limited to usability studies in artificial conditions (i.e. usability labs) with users that have

http://www.welie.com/
never experienced the patterns in real projects. Moreover, quite often the participants are students and/or researches which are not representative of a living experience with design patterns in the industry.

In order to understand how design patterns are actually used by development teams in the industry, we propose an empirical evaluation of a pattern language in a company. The general research question was to understand how the pattern language is currently used in the enterprise. By conducting expert interviews combined with a kind of filed usability study (including the usage of the language patterns to solve problems by the experts) we wanted to investigate how pattern languages are currently used by analysts in a real application context.

3. THE ENTERPRISE AND ITS PATTERN LANGUAGE

This study was carried out at SmaIS-MvM, a non-profit organization devoted to the design, deployment and handling of public e-government applications in Belgium. SmaIS-MvM started developing a pattern language as a means to support the discussion of solutions with stakeholders (many and with different background), which should finally lead to the standardization of the user interface of the web-based applications produced by different teams in the organization. Such a pattern language was implemented in October 2006 as a result of a two years project of the usability department. Several seminars were organized to teach stakeholders how to use the patterns language in their activities. The evaluation of the pattern language in terms of usability and integrated usage in the development process was conducted in April 2008.

The UI design patterns have been identified by browsing existing applications designed by SmaIS-MvM among the ones already deployed or at advanced development stages. The patterns are organized in three different levels of UI granularity covering the so-called screen flow patterns (e.g. sequence of steps to accomplish an administrative procedure), web page patterns (e.g. page layout), and basic components (e.g. form fields). The description of UI patterns is rather classical (advices of implementation and rationale around a given UI design problem) but it might also include wireframes for supporting low-level fidelity UI prototyping using design patterns. According to the level of granularity of the UI patterns, the wireframe consists of a schematic representation of the layout and disposition of an UI element (e.g. a page or a form), or in a rough schema of the navigation. In order to provide a non-ambiguous description of navigation, navigation between pages is described using a formal description technique called StateWebCharts (SWC) (Winckler and Palanque, 2003). Figure 1 shows the pattern called “Multi-Step Wizard”. It exemplifies the use of the wireframe description using SWC (see the WIREFRAME field in Figure 1) for describing the sequence of pages at the screen flow level. The wireframe section presented in Figure 1 provides three alternative solutions for implementing that pattern (i.e. sub-patterns: strong guidance wizard, supple guidance wizard, and editable summary). It is also noteworthy that the pattern description provides explicitly important information leading to the implementation such as points to standard user interfaces components (including MS Visio templates\(^3\), java script menus, style sheets, etc) that should be integrated into design as part of the solution. The set of patterns is available on the web with a connection to the framework Eclipse. Patterns are interlinked in order to ensure internal consistency between related solutions (e.g. patterns describing page forms are connected to patterns describing form fields).

The pattern language is used by five user groups within SmaIS-MvM: the group of analysts doing the (function) specification of projects, user interface developers, designers, content editors and a group of people directly responsible for the implementation and maintenance of the pattern language. The majority of tasks for the five user groups are related to consulting the pattern language either in terms of looking up wireframe structures or code fragments, or to contribute to the pattern language either by commenting or proposing additional content. The group of responsible persons for the pattern language additionally performs task in terms of set up and changing the content of the pattern language.

---

\(^3\) MS Visio: http://office.microsoft.com/visio/
MULTI-STEP WIZARD

WHAT The goal of the procedure is reached through the accomplishment of a sequence of activities. This sequence of activities is guided by the sequence of screens but also by the navigation proposed which is limited to “next step” and “previous step” (eventually “cancel all”).

WHEN AND HOW TO USE Should be used when users are novice
Shouldn’t be used when the procedure is long & likely to be interrupted

LAYOUT 1) Distinguish procedure steps (ex. Step 2) and auxiliary pages (ex. OVERVIEW page)
2) see WIZARD STEP pattern for the layout of each step

COPY Give the procedure a clear title, whose formulation is user-centred and contains a verb corresponding to the goal of the procedure.

IMPLEMENTATION 1) Make sure the back button always works.
2) When going back to a previous step, auto complete the previous input

RESOURCES 1) http://www.designofsites.com/about_the_book/pattern1.pdf
2) http://harbinger.sims.berkeley.edu/ui_designpatterns/webpattern2/
webpatterns/pattern.php?id=7

WIREFRAME Several implementations are possible, just around the way provided for the edition of the overview page. See MULTI-STEP WIZARD sub patterns to identify which one fits to your situation.

SUB PATTERN 1) Strong guidance wizard

SUB PATTERN 2) Supple guidance wizard

SUB PATTERN 3) Editable summary

Figure 1. Example of UI analysis pattern at the Screen Flow level: “MULTI-STEP WIZARD”

4. METHOD, PARTICIPANTS AND PROCEDURE

The goal of the evaluation was to investigate if the pattern language is usable for problem solving in the analysis phase including the usage of wireframes and code segments. Are the patterns understood? Are people using patterns (in their daily work) and how do they solve problems using these patterns? Is the usage of the pattern language only kind of automatic (following the suggestions) or are the users involved in the pattern language and could contribute to the pattern language (e.g. by adding new forms of patterns)? Three working hypothesis guided the evaluation:

1. The pattern language in place and the used description formalism in hypertext are usable and have a satisfying level of detail to support the concept stage of the user interface.
2. The wireframe and the hypertext structures are helpful to reach consensus for the user interface solutions in the project.
3. The pattern language helps to support the reuse of solutions made during the conception of user interfaces.

Goal was to investigate in general the usage of the pattern language in terms of problem solution (understanding and finding the patterns necessary for solving a problem) and to investigate how deep the understanding of the pattern is (taking the possibility to contribute/comment on patterns or even to develop new patterns as a measurement for understanding of the patterns).
Eight participants were taking part in the interview (six male, two female), age ranged from 24 to 31 (mean: 26 years, SD 2.5, n=7). Five participants indicated to work mainly as analysts; two indicated to be mainly leading projects, and one was mainly working as a system architect. Three participants were working more than three years at SmalS-MvM, two participants up to two years and three participants were at SmalS-MvM for 3, 6 and 9 months respectively. The pattern language was used in projects by six participants; two participants did not use the pattern language within their projects. One participant reported not to use the pattern language, six mentioned to look up solutions for specific problems, one dropped comments to improve the pattern language, four used personalized wireframes and five used pattern language to argue for solutions of the user interface in projects.

The evaluation was conducted at SmalS-MvM using an expert interview combined with having the experts conduct some tasks to reflect on their actual usage. Participants were asked to fill out a pre-questionnaire, following they were asked to conduct three tasks at their workplace, followed by a final interview including questions about possible improvements for the pattern language. The three tasks were related to investigate the working hypothesis, on how usable the pattern language is to solve a typical user interface problem at concept stage, if the people understand the limitations of the pattern language in terms of necessary modification and if people can add patterns, based on solutions they might develop in projects. All tasks were related to the following problem: Goal is to set-up a webpage for the lunch distribution of a school with a personal identification for parents and the possibility to select the weekday lunch is taken by the kids. Parents should have the possibility to cancel lunch three days in advance and the school staff to look up the number of ordered lunches.

Part one of task 1 (T1/a) was to find a general solution for the lunch webpage, and it intended to trigger a solution using the pattern language by selecting a wireframe page. Part two of task one was to produce an html page, incorporating the solution (T1/b). Task 2 was to compare the solution of the participant with the proposed pattern language solutions (in case the proposed pattern was not used in task 1), and how to modify the pattern language if necessary. Task 3 was to develop a solution for paying the lunch fee by credit card. The pattern language did only partly contain patterns for this type of problem, thus it should evoke the need to add patterns to the language.

5. RESULTS

The results of the expert interviews were transcribed and analyzed using the conducted tasks as a means of structuring the answer categories (see also Table 1). The results are first grouped based on the tasks conducted and then we answer the three working hypothesis.

Task one was set-up to understand if the pattern language is usable within the concept phase. From the eight participants performing the task four were choosing the pattern language to help during the development of their solution. The other four participants did not look up the pattern language for any form of guidance. When prompted by the leader of the test to use the pattern language to set-up wireframe structure and following the html page, the usage of the pattern language was still low for the following tasks. Overall six out of eight participants looked up less than four pages in the pattern language for all three tasks. Only two out of eight participants looked up more than four pages. Task one was central to understand if the pattern language is usable. It took participants on average 17.5 min (ranging from 6 to 35 minutes to complete the task). Five participants were using the pattern language, looking at 2.6 pages on average. Fastest completion of the task was with using a similar (existing) application (not the pattern language). Using patterns to set-up a first HTML page was taking about 6 min on average. All participants were knowledgeable in using state web chart (SWC), but solutions did not include the restriction for modification (parents can not cancel later than three days before).

Task two on finding modifications or changes based on the pattern language took on average about 8 min. The presentation of patterns and associated patterns is leading to a reduced usability of the pattern language. Patterns were difficult to find. Task three was performed as an expert interview. Participants were asked how they would solve the task and what kind of modifications would be necessary to improve the pattern language. Table 2 gives an overview on the results of the various aspects prompted in the expert interviews. We summarize these results showing the usage of the patterns and experts expressed opinions about user satisfaction.
Table 1. Results from the investigated and mentioned topics for each task: Frequencies of usage aspects based on the conducted tasks (indicated by (*)) and results from the expert interviews (indicated by (**)).

<table>
<thead>
<tr>
<th>Task</th>
<th>Investigate Topic</th>
<th>Pattern language Usage (*)</th>
<th>Integration in MS Visio (*)</th>
<th>User Satisfaction (**)</th>
<th>Adequacy of Pattern language (**)</th>
<th>Method Take-up (**)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1/a: Designing with Wireframe</td>
<td></td>
<td>++</td>
<td>++</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td>T1/b: Designing with SWC</td>
<td>Knowledge of SWC (***)</td>
<td>Hypertext Modeling (*)</td>
<td>Usage of SWC (*)</td>
<td>User Satisfaction with SWC (****)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2: Discussing alternatives</td>
<td>Readability (**)</td>
<td>Usability culture (***)</td>
<td>Adaptability of the Pattern (****)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3: Updating (because pattern is missing (*))</td>
<td>Resources Used (*)</td>
<td>Agility (*)</td>
<td>Gratitude (**)</td>
<td>Motivation (**)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: ++++ very frequently, ++ frequently, + sometimes, + rarely, - never | SWC = state web chart.

For the hypothesis we can summarize the results from the expert interviews as follows:

**H1: The Pattern language is usable and helps to find solutions within the concept phase.**

The usability of the pattern language is acceptable but leaves room for improvement:
- **Navigation View:**
  The Pattern language enables each group to use a different navigation view, to better support the user groups at the various development stages. Subjects using the Pattern language more frequently reported for example, that they typically only look at the analysts view or the designers view, but they would not look up the technical view (with all the technical details for the implementation of the html page).
- **Navigating Guidelines:**
  During the execution of the tasks the pattern language was rarely used, the Visio integrated parts were hardly known by the users, but in general the users reported to be satisfied with the Pattern language, and the method was adequate. The scenario for developing the html code (1/b) showed that State Charts are known by the user, patterns are used for the hypertext but the usage is not optimal (only some of the users use the support).
- **The integration of the pattern language into Visio poses different usability problems in terms of access, understanding or usability problems that infer the usage of the Pattern language but originate from users not knowledgeable in Visio.**

**H2: The wireframe and the hypertext structures are helpful to reach consensus for the user interface solutions in the project.**

The Pattern language itself is well perceived in terms of usefulness, it is valued in terms of readability and that it can be adapted and modified if necessary. The general usability culture of the enterprise shows that the patterns are used to reach consensus in discussion and during project development.

**H3: The pattern language helps to support the reuse of solutions made during the conception of user interfaces.**

The pattern language is helping to re-use solutions made during the analysis phase, but limited time in the projects do not support the usability culture of the enterprise. Problematic in the usage of the Pattern language is the missing knowledge of what patterns/solutions are already available and what patterns are missing.

The success in a real application domain will depend on the ability to keep the language a living document. E.g. Participant 4 was suggesting “I feel concerned by the evolution of the pattern language, because if it [the pattern language] does not evolve it will become unusable soon”. Of course the development of the Pattern language will be affected by time constraints stemming from the various projects the analysts are involved in. Participant 6 stated “I understand the importance of this issue [about the evolution of the pattern language], however project deadline is our priority”.
The analysts working with the language seem in general quite motivated to contribute to improvements with the project. Participant 8 stated: “I and one of my colleagues have contributed to this [pattern language] project, we have create a special kind of document called ‘requirements technique’ for defining [our] responsibilities during the design of the user interface.”. However, as stated by the same participant, quite often they do not know if they are dealing with general patterns or just local solution to problems: ‘... I cannot say [the document requirement technique] it is actual standard inside SmalS [outside my development team]. Anyway, our goal was to provide a solution to [recurrent problems] we have encountered while creating relationships between the database and the screen shoots [present in design patterns] to describe the data flow sequencing and their relationship with screen flows, so that we could have a better understanding on how to fill in the forms”.

From the perspective of a general usability culture the pattern language is welcome within the group of analysts. In general participants were well aware of the pattern language, with the limitation that some participants were not aware of the latest version (including a number of essential improvements). In terms of modifications most analysts are willing to contribute to the language, but feel restricted in doing so by time and budget constraints in their projects. Other analysts reported that they do not see their role in modifying the content but they would prefer the usability department to update the language, based on a more general consensus of what kind of patterns would be useful to add.

6. LESSONS LEARNED AND INSIGHTS

The contribution of the present work is two-fold: first the qualitative study allows us to understand how a pattern language is used in industry and to what level and degree analysts use the pattern language. Second, the results of the study allow us to compare theoretical claims on the use of patterns and their actual use in the real world. This study demonstrates many of difficulties one can encounter when evaluating pattern languages, in particularly due to the absence of sound quality models for describing pattern languages.

Based on the evaluation it is clear that supporting user-centered development with user interface patterns is depending on the usability culture within the enterprise, on the modifiability of the pattern language, on the available resources to allow the pattern language to be a living document. The following conclusions and recommendations can be drawn from the qualitative analysis of the interviews and the three scenarios conducted. The pattern language usage should be improved by:

- Encouraging the participation of users in the development and modification of the pattern language;
- Improving the usability especially the navigation within the pattern language including the state web charts depiction (maybe including how the final UI would look like);
- Enhancing the communication and “advertising” of the availability of the pattern language as well as of modifications of the pattern language to all possible users in the enterprise; and allow evolvement of the language, also in terms of time and budget resources.

The pattern language used at SmalS-MvM is a useful means during the development of applications in the analysis phase. The major benefit for the user is the ability to focus on the (technical and) functional specification of the interface and to be able to discuss on a functional basis the interface elements with clients. We could observe that stakeholders at the organization use the patterns for similar purposes. However different sections of the patterns are perceived with varying levels of importance by the users. Due to several constraints, we had to focus our study on just one category of stakeholders (i.e. the analysts) and their activity in a specific design phase of the development process (i.e. design of prototypes). However, we suggest that further investigation should be performed to understand if the pattern language available fulfills other stakeholders’ needs too. Moreover, it would be interesting to observe the interaction of different stakeholders around the pattern language at different stages of the development process.

The usage of the presented pattern language was limited in the enterprise. Experts reported that the guidelines could not be easily navigated, integration into Visio was perceived as difficult and especially the missing modifiability of the patterns and the understanding of what the pattern language has to offer in terms of solutions, made the language difficult to use. At this point we can question if empowering the development team with ready-made solutions to implement application from patterns (for example using MS Visio components) is really improving the usability culture in the enterprise.
Generally speaking, this study shows an agreement between research and usage in practice. Researchers and practitioners are keen to adopt and develop pattern languages, however, it seems that some methods are missing to guide the identification of emerging patterns and to judge the validity of the description provided. High quality of pattern languages is a pre-condition for large adoption of pattern in projects. Patterns should be validated to improve the trust in the pattern language. Based on observations made at SmalS-MvM we can state that pattern languages are a good solution to leverage the usability culture into organizations, but pattern languages should evolve to meet organization needs (in particular stakeholders’ needs for their projects). Otherwise the pattern language will have its credibility compromised and lose the added value as a tool for supporting the communication of usability issues and for generating ready-made solutions.

REFERENCES

Gamma, E.et al. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.