

MOUDIL: A Comprehensive Framework for Disseminating and Sharing HCI Patterns

A. Gaffar, D. Sinnig, H. Javahery, A. Seffah

Human-Centered Software Engineering Group
Concordia University, Montreal, QC, Canada, H3G 1M8
{gaffar, d_sinnig, h_javahe, seffah}@cs.concordia.ca

Keywords

HCI patterns, pattern languages, Integrated Pattern Environment, tools for patterns

INTRODUCTION

In the area of user interface (UI) and usability engineering, patterns have been introduced as a medium to capture and disseminate user' experiences and design practices/knowledge. They are a vehicle for transferring, by means of software development tools, the design expertise of human-factor experts and UI designers to software engineers, who are usually unfamiliar with UI design and usability principles. However, the lack of common fulcrum and central repository for patterns make it hard to achieve this goal. Several pattern writers have introduced their own terminology and ontology of patterns. In this paper, we will introduce our methodology, which will highlight a path through the heterogeneous pattern world, and result in a comprehensive framework for disseminating and sharing HCI patterns.

In addition to the above, a further challenge is the lack of tool support, which makes it difficult to capture, disseminate and apply patterns effectively and efficiently [7]. Pattern writers, who are most often usability engineers with a background in psychology and cognitive science, must try to convey complex information to describe the user problem and design solutions in a comprehensible and comfortable way. Pattern users, who are most often software developers unfamiliar with usability engineering techniques, need also tools to understand when a pattern can be applied (context), how it works (solution), why it works (rationale), and how it should be implemented.

To address this problem, our team is developing on MOUDIL (Montreal Online Usability Patterns Digital Library) which is a method + a tool for capturing and disseminating patterns. This paper is an overview on MOUDIL.

THE SEVEN C'S METHODOLOGY

Like the software engineering community, the user interface design community has been a forum for vigorous discussion on patterns. Many groups have devoted themselves to the development of pattern languages for user interface design and usability. Among the heterogeneous collections of patterns, *Common Ground* [9], *Experiences* [3], *Brighton* [5] and *Amsterdam* [10] play a major role in this field and wield significant influence. However, each of these collections introduces its own terminology, classification system and notation/format [8]. Our methodology called "The seven C's" aims of centralizing and organizing patterns into one repository, as well as disseminating pattern knowledge to the HCI community. Our methodology is as follows:

Collect: Place Different Research Work on Patterns in One Central Data Repository

Numerous works on patterns have been developed in the HCI community, however they are scattered in different places. A central repository of patterns will allow the user to concentrate on knowledge retrieval, rather than wasting time on searching for patterns. For this reason, we are collecting known references on patterns into one corpus. Currently our corpus includes more than 300 patters.

Cleanup: Change from Different Formats/Presentations into One Style

Ideally, different works on patterns deal with different problems. However, as we went through step 1, we were able to identify that some patterns are dealing with different sides of the same problem (correlated patterns), some patterns are offering different solutions to the same problem (peer patterns/competitors) and some are even presenting the same solution to the same problem (similar), only in different collections with different presentation

formats (redundant patterns). Since a large number of patterns have differing presentation formats, it is difficult to detect these and their relations with other patterns.

Putting patterns in a unified format will help discover these relationships, put related patterns closer together, and possibly remove the redundancies/inconsistencies. This is an extremely important for building a common ground. We are conducting further research to improve the presentation format and change it from an *art*, requiring a lot of creativity and expertise, to a systematic and possibly automated approach. This will be developed further in the *Control* step.

Certify: Define a Domain and Clear Terminology for Our Collection

The available work on patterns is tremendous; it is not wise, feasible or even useful to collect everything in one place. To make any collection useful, it has to focus on a specific domain. For this reason we are working on terminology to clearly define what belongs in our collection to make it inclusive and concise.

Contribute: Receive Input from Pattern Community

New patterns emerge all the time in all areas of the scientific community, including HCI. It is very difficult to keep track of these emerging patterns. Typically, it would take years before an expert can come up with a thorough collection of patterns [1, 2, 4] or have time to update an existing collection [9, 10]. Having one central repository for patterns will help to unify pattern knowledge captured by different individuals. Furthermore, such a repository will help to add emerging patterns quickly, so that they are made available to the community. We will therefore have a continuously evolving collection of patterns.

Categorize: Define Clear Categories for Our Collection

Within our collection, we need to be able to create a hierarchy of categories to make them manageable. The first goal of categorization is to reduce the complexity of searching for, or understanding, the relationship between patterns. The second, and more important goal, is to build a model for our categories. We are inspired by the evolution process in other domains like C++ (hierarchies in I/O classes, STL hierarchies, etc) and Java (evolving hierarchies in event handling models, etc).

Connect: The Second Level of Complexity – Establishing Semantic Relationships between Patterns in a Relationship Model

A significant part of knowledge associated with patterns lies in the relationships between them. Finding and documenting these relationships will allow developers to easily use patterns as one integrated part to develop an application, instead of relying on their common sense and instinct to pick up some patterns that seem to be suitable. A proven model for the pattern collection will help to define an ontology for the pattern research area with all proper

relationships such as inference, equivalence and subsumption between patterns.

Control – Machine Readable Format for Future Tools

Once a model is established, it will enhance the process of automating the UI design. The ultimate goal of many applications is to interact with the machine as a viable partner that can read, understand our work, and then contribute to it in an intelligent way. In short, having a machine-readable format can help automate the process of UI design using patterns.

INTEGRATED PATTERN ENVIRONMENT – MOUDIL

In order to accelerate the implementation of our seven C's methodology, tool and software support are necessary. Each step requires a certain tool. Our research team is currently developing an Integrated Pattern Environment (IPE), called MOUDIL, which will unite all necessary functionality. MOUDIL was originally designed with two major objectives: Firstly, as a service to UI designers and software engineers for UI development. Secondly, as a research forum for understanding how patterns are really discovered, validated, used and perceived. One last objective for MOUDIL, in addition to the above two, is to use it as a prototypical implementation of an IPE. It will be able to provide functionality that supports every step of the seven C's methodology.

In particular, MOUDIL provides the following key features:

- MOUDIL has been designed to accept proposed or potential patterns in many different formats or notations. Therefore patterns in versatile formats can be submitted for reviewing.
- An international editorial board for reviewing and validating patterns. Before publishing, collected and contributed patterns must be accessed and acknowledged by the editorial committee. We are inviting HCI patterns practitioners and researchers to join this committee.
- Pattern Ontology editor captures our understanding of pattern concepts and puts them into relation with each other (Taxonomy).
- The MOUDIL Pattern Editor allows us to attach semantic information to the patterns. Based on this information and our ontology, patterns will be placed in relationships, grouped, categorized and displayed.
- The pattern navigator provides different ways to navigate through patterns or to locate a specific pattern. The pattern catalogue can be browsed by pattern groups or searched by keyword. Moreover, a pattern wizard will find particular patterns by questioning the user.
- The pattern viewer provides different views of the pattern, adjusted to the preferences of the specific user.

CONCLUSION AND FUTURE AVENUES

HCI patterns are useful in gathering and documenting experiences for future developers. A great deal of work has been done on HCI patterns by many different individuals. The lack of knowledge centralization, however, requires users to hunt for suitable patterns, and extract them for their own use. Gathering relevant patterns in one repository will help overcome this difficulty.

In our future research, we want to use the gathered information from the collected and analyzed patterns to come up with a formal pattern notation. Such a notation will help capture and disseminate pattern knowledge effectively.

REFERENCES

1. Alexander, C. *The Timeless Way of Building*. New York: Oxford University Press, 1979.
2. Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I., and Angel S., *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977.
3. Coram, T., Lee, J. *Experiences - A Pattern Language for User Interface Design*. Available at <http://www.maplefish.com/todd/papers/experiences/Experiences.html>.
4. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
5. Griffiths, R. *Brighton Usability Pattern Collection*. Available at <http://www.it.bton.ac.uk/cil/usability/patterns/>.
6. HCI Department, Concordia University, Montreal. *MOUDIL: Montreal Online Usability Digital Library*. Available at <http://hci.cs.concordia.ca/moudil/homepage.php>.
7. Seffah, A., Javahery, H. *A Model for Usability Pattern-Oriented Design*. In Proceedings of TAMODIA 2002, (Bucharest, Romania, July 2002).
8. Seffah, A., Javahery, H. *On the Usability of Usability Patterns*. Workshop entitled *Patterns in Practice*, CHI 2002, (Minneapolis, Mi, April 2002).
9. Tidwell, J. *COMMON GROUND: A Pattern Language for Human-Computer Interface Design*. Available at http://www.mit.edu/~jtidwell/interaction_patterns.html
10. Welie, M. *Interaction Design Patterns*. Available at <http://www.welie.com/patterns/>.