Pattern-oriented UI design based on user experiences: A method supported by empirical evidence
Javahery, Homa
ProQuest Dissertations and Theses; 2007; ProQuest
pg. n/a

Pattern-Oriented UI Design Based on User Experiences: A Method Supported by Empirical Evidence

Homa Javahery

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

December 2006

© Homa Javahery, 2006

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Library and Archives Canada
Bibliothèque et Archives Canada

Published Heritage Branch
Direction du Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-30131-9
Our file Notre référence
ISBN: 978-0-494-30131-9

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

Pattern-Oriented UI Design Based on User Experiences:
A Method Supported by Empirical Evidence

Homa Javahery, Ph.D.
Concordia University, 2006

User-Centered Design (UCD) is a philosophy surrounding interactive system design, with the purpose of achieving product usability. One challenge with UCD and its related methods is the lack of a concrete process which supports designers in building user interface (UI) designs founded on user experiences. In current practice, design decisions are made based on loosely-defined guidelines, giving rise to a significant "gap" between user analysis and design outcomes. This is especially problematic for novice designers who lack the background and training required to make trade-offs, judgments and interpretations towards a usable design.

In this thesis, we propose a Pattern-Oriented UI Design method which is driven by user experiences. It is founded on a set of core UCD principles which we have enriched with "engineering-like" concepts such as reuse and traceability. The method is based on two key artifacts – personas, used to model user experiences, and patterns, used to capture best design practices. Following this method, we define the *UX-P Process*, a systematic process which is semi-automated and characterized by rigorously-defined steps; designers iteratively create personas, select patterns, and compose patterns into a comprehensive design, based on user specifications and usability considerations. We have built a supporting tool, which allows designers to cluster users into personas and select candidate patterns based on persona specifications.

We carried out two empirical studies with end-users. The goal of the first study was to assess the feasibility of the method; the second, to validate the process. Both studies were carried out with Bioinformatics applications and were comparative in nature testing the original design with our prototype. The outcome of these empirical studies indicated a positive increase in usability measures for our design prototypes, including a significant improvement in task times and user satisfaction.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ACKNOWLEDGEMENTS

I am grateful to my co-supervisors, Dr. Ahmed Seffah and Dr. T. Radhakrishnan, for their guidance and direction throughout my doctoral research. I thank my committee members, Dr. Volker Haarslev, Dr. Olga Ormandjieva, and Dr. Yong Zeng, for their sound judgment and feedback.

I acknowledge NSERC (Natural Sciences and Engineering Council of Canada) and FQRNT (Le Fonds Québécois de la Recherche sur la Nature et les Technologies) for their financial support.

I am very thankful to all the study participants: End-users, domain experts and HCI experts. Without them, this research work would not be possible.

I am indebted to Daniel Sinnig, my colleague and friend, who spent hours brainstorming with me about my approach. He truly made my years as a graduate student enjoyable and rewarding, and gave me that extra push when I needed it the most.

I would like to express my gratitude to the following colleagues: Alexander Deichman, for his involvement in this project and for giving me a different perspective. Eric Waddell, for his software development expertise. Stephen Barrett, Rozita Naghshin and Maik Wurdel, for their feedback.

I thank Behzad Hadian, for his unrelenting support. I thank him for giving me the hope and encouragement to continue, and for believing in me.

Above all, I am grateful to my family. I thank my sister, Sima, for her active support throughout these past few years. I thank my parents for showing me the meaning of dedication, integrity and hard work.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

Chapter 1 Introduction 1

1.1. Motivation 1
1.2. Research Statement and Objectives 2
1.3. Terminology 5
1.4. Organization of the Thesis 6

Chapter 2 Background and Related Work 9

2.1. UCD and Interactive Design Methods 9
2.2. Modeling Users and their Experiences 12
2.2.1. Cognitive Styles 12
2.2.2. User Profiles 13
2.2.3. Personas 14
2.3. Building Conceptual Designs 20
2.3.1. Guidelines 24
2.3.2. Claims 25
2.4. Patterns and Pattern-Oriented Design 26
2.4.1. Original Ideas about Patterns in Design 26
2.4.2. HCI Patterns 27
2.4.3. HCI Pattern Languages 30
2.4.4. Patterns and the User Interface Design Process 33
2.5. Linking User Experiences and Design Solutions 37
2.5.1. Goal-directed design 37
2.5.2. Welie's Layered Model of Usability 39
2.6. Summary of Related Work 41

Chapter 3 Designing with Personas and Patterns: Case Study 1 43

3.1. Framework 43
3.2. Applying the Persona to Pattern Framework 45
3.2.1. Understanding the Domain and Users 45
3.2.2. Initial Persona Set 46
3.2.3. Pre-design Usability Evaluation 48
3.2.4. Refined Persona Set 52
3.2.5. Selecting Patterns based on Personas 52
3.2.6. Composing Patterns 54
3.3. Post-Design Usability Evaluation 56
3.3.1. Participants 56
3.3.2. Method 57
3.3.3. Results 61
3.4. Summary of Case Study 1 65

Chapter 4 Proposed Design Method 67

4.1. Overview 67
4.2. Key phases 69
4.2.1. Persona creation 69
4.2.2. Pattern selection 71
4.2.3. Pattern composition 73
4.3. Key principles 74

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.3.1. Focus on all user groups and their needs ... 74
4.3.2. Incorporation of behavioral rationale ... 74
4.3.3. Systematic and traceable application of knowledge ... 75
4.3.4. Embracing Reuse ... 76
4.3.5. Light weight and pragmatic approach ... 76
4.4. Summary of Proposed Design Method ... 78

Chapter 5 Personas and Patterns: Process Requirements ... 79

5.1. Overview ... 79
5.2. Persona Model ... 81
5.3. Pattern Model ... 84
5.4. Heuristics for Creating Personas ... 86
5.5. Criteria for Selecting Patterns ... 88
5.5.1. Documenting the Selection Rationale ... 88
5.5.2. Assimilating Usability Principles ... 91
5.5.3. Partitioning Usability Principles: Needs and Pattern Criteria ... 93
5.5.4. Relating Usability Needs to Pattern Criteria ... 95
5.5.5. Relating User Variables to Usability Needs ... 96
5.5.6. Summarizing the Dependencies ... 98
5.6. Summary of Process Requirements ... 101

Chapter 6 The UX-P Process ... 102

6.1. Process Overview ... 102
6.2. Persona Creation ... 106
6.2.1. Clustering users [Step 1 in Fig. 6-2] ... 106
6.2.2. Modifying clustering parameters [Step 1b in Fig. 6-2] ... 110
6.2.3. Refinement of Persona Set [Step 2 in Fig. 6-2] ... 112
6.3. Pattern Selection ... 116
6.3.1. Rule-based scoring technique ... 116
6.3.2. Selecting Patterns based on Personas [Step 3 in Fig. 6-2] ... 119
6.3.3. Modifying selection parameters [Step 3b in Fig. 6-2] ... 126
6.3.4. Filtering pattern set [Step 4 in Fig. 6-2] ... 128
6.4. Pattern Composition ... 128
6.4.1. Composing patterns [Step 5 in Fig. 6-2] ... 129
6.5. Summary of the UX-P Process ... 141

Chapter 7 Operationalization and Validation ... 142

7.1. Tool Support: P2P Mapper ... 142
7.2. Validation: Case Study 2 ... 145
7.2.1. Domain and Users ... 145
7.2.2. Applying the UX-P Process ... 148
7.2.3. Testing the Prototype with End-Users ... 155
7.3. Summary of Case Study 2 ... 159

Chapter 8 Conclusion ... 160

References ... 165

Appendix A: Interactive Design Methods and Limitations ... 172

Appendix B: The UPADE Web Language ... 176

Appendix C: Pre-Design Usability Evaluation of NCBI Site ... 177

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix D: Task-Based Evaluations 181
Appendix F: Rule-based scoring technique 188
Appendix G: Persona Example for Protein Explorer 190
Appendix H: Empirical Studies in HCI 192
Appendix I: Process Described in Pseudo-code 198

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF FIGURES

Figure 1-1. Current Problem of Deriving a Design from User Experiences 3
Figure 2-1: Constituents of a UI Conceptual Design (adapted from Baxley [2003]) 21
Figure 2-2: Basic Conceptual Design of a Website 21
Figure 2-3: Conceptual and Detailed Designs of a Website [Najjar 2000] 22
Figure 2-4: Conceptual and Detailed Designs of a Visualization Application 23
Figure 2-5: The Experiences Pattern Language 32
Figure 2-6: The Pattern-Supported Approach Framework 35
Figure 2-7: The Goal-Directed Design Process [Cooper and Reimann 2003] 37
Figure 2-8: Layered Model of Usability [Welie et al. 1999] 39
Figure 3-1: Persona to Pattern Framework 44
Figure 3-2: Satisfaction ratings of novice and expert users of the NCBI Web site 51
Figure 3-3: Pattern Skeleton of the NCBI Homepage 55
Figure 3-4: Pattern-Oriented Design of NCBI homepage 55
Figure 3-5: Testing Protocol for NCBI Study Post-Design Usability Evaluation 58
Figure 3-6: Morae Usability Testing Tool 59
Figure 4-1: Proposed Design Method 68
Figure 5-1: Requirements for UX-P Process 80
Figure 5-2: Criteria for selecting patterns 90
Figure 5-3: Usability-design model for pattern selection 91
Figure 5-4: Dependencies between selection components 98
Figure 6-1: Black Box View of the UX-P Process 102
Figure 6-2: Process Diagram for UX-P Process 104
Figure 6-3: Example of rule-based scoring technique 123
Figure 6-4: Comparison of Similar Patterns 130
Figure 6-5: Two Competitor Patterns 131
Figure 6-6: Home Page pattern with sub-ordinate patterns 132
Figure 6-7: Sequential Pattern 133
Figure 6-8: Hierarchical Pattern 133
Figure 6-9: Comparison of Stack and Tiled Page Patterns 135
Figure 6-10: Use of Navigation Page Pattern in [Montreal City 2006] 136
Figure 6-11: Index Browsing and Executive Summary Patterns 139
Figure 6-12: Use of Global Navigation pattern in [Montreal City 2006] 140
Figure 7-1: Overview of P2P Mapper Tool 144
Figure 7-2: Protein Explorer Application 146
Figure 7-3: Task Model for 3D Visualization Tool 147
Figure 7-4: Clustering Tool in P2P Mapper Environment 150
Figure 7-5: P2P Mapper view of the Persona Marta 152
Figure 7-6: Protein Explorer prototype designed using UX-P Process (first set of patterns) 154
Figure 7-7: Protein Explorer prototype designed using UX-P Process (second set of patterns) . 154
Figure B-1. An Overview of the UPADE Web Language 176
Figure C-1. Original and redesigned NCBI homepage using patterns 179
Figure E-1: Homepage for general public 187
Figure E-2: Homepage for universal access 187

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF TABLES

Table 2-1: Interactive Design Methods and Limitations 11
Table 2-2: Persona Components (adapted from Courage and Baxter [2005]) 20
Table 2-3: Example of a Claim 25
Table 2-4: HCI Pattern for Visualization Environments 30
Table 2-5: Pattern Groups Ordered According to a Web Design Process [van Duyne 2003] 35
Table 3-1: First Iteration of Persona Set 47
Table 3-2: Selected User Characteristics for Pre-Design Evaluation of NCBI Site 49
Table 3-3: Task Use and Interaction Behavior of NCBI Site Users 50
Table 3-4: Selected Information from Personas and Associated Patterns for Study 1 53
Table 3-5: Aggregate description of novice participants 56
Table 3-6: Aggregate description of expert participants 56
Table 3-7: Background of novice users 57
Table 3-8: F values of ANOVA single factor tests when comparing task times 62
Table 3-9: F values of ANOVA single factor tests when comparing failure rates 62
Table 4-1: Wizard Pattern for Users with Guidance Needs [Welie 2003] 72
Table 5-1: Knowledge Elicitation Techniques Used 80
Table 5-2: UX-P Process Requirements and Elicitation Techniques Used 81
Table 5-3: Persona Elements 81
Table 5-4: User Variables in the Persona Model 83
Table 5-5: Pattern Variables 85
Table 5-6: Pattern model elements 86
Table 5-7: Examples of Bastien and Scapin's [1993] ergonomic criteria 92
Table 5-8: Assimilated Usability Principles from Various Sources 93
Table 5-9: Usability needs 94
Table 5-10: Pattern Criteria 94
Table 5-11: Dependencies between usability needs and pattern criteria 95
Table 5-12: Dependencies between user variables and usability needs 96
Table 5-13: Types of relationships identified for a user variable V and a usability need N 97
Table 6-1: Representation of the user Chantal with persona model elements 108
Table 6-2: Current mobile phone users with gaming subscriptions 109
Table 6-3: User clusters 109
Table 6-4: Second iteration of user clusters 111
Table 6-5: Persona skeletons 113
Table 6-6: Populating user variables for the persona Anna 114
Table 6-7: Narrative View of the Persona Anna 115
Table 6-8: Rules for associating user variables to usability needs 118
Table 6-9: Examples of patterns and related pattern criteria 118
Table 6-10: User variables of the Persona Serge 120
Table 6-11: Narrative view of the Persona Serge, a low-literate user 122
Table 6-12: A subset of recommended patterns 124
Table 6-13: A modified subset of patterns with scores 127
Table 6-14: Alternative Orthography Pattern 138
Table 7-1: UX-P Process Steps and Tool Support 142
Table 7-2: Aggregate description of 22 participants of the study 148
Table 7-3: Results after Modification of Clustering Parameters 151
Table 7-4: User Variables for Pattern Selection 153
Table 7-5: Aggregate description of 15 participants of the study 155
Table 7-6: Results for task duration 157

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 7-7: Results for failure rate 158
Table C-1: Subset of Heuristics [adapted from Nielsen 1994] 179
Table E-1. Simplified Text Pattern 184
Table E-2. Alternative Orthography Pattern 185
Table E-3. Audio Communication Pattern 186
Table E-4. Original and redesigned NCBI homepage using patterns. 187
Table E-5. Original and redesigned NCBI homepage using patterns 187
Table F-1. Rules for associating user variables to usability needs 189
Table F-2: Associating user needs to pattern criteria 189
Table G-1. Persona 1 for Protein Explorer 191
Table H-1: Experimental Assessment Summary for NCBI Study 194
Table H-2: Experimental Assessment Summary for Protein Explorer Study 195
Table H-3: Usability Evaluation Methods [Ivory and Hearst 2001] 196

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 1

Introduction

In this chapter we introduce the motivation for this research work, define the research statement and objectives, and clarify some key terms. We also give an overview of the thesis organization.

1.1. Motivation

Interactive systems are increasingly entwined in our daily lives. Whether we are dealing with the design of conventional desktop, web, and mobile applications, or even wearable and immersive computing platforms, it is apparent that a plethora of new challenges exist. A large part of most interactive systems is the user interface (UI) component. It has been reported that the UI component represents more than 50% of the code and more than 60% of total development costs [Myers 1995]. In addition to available technology and hardware considerations, designers need to be aware of the many different factors which influence UI design, including users and their characteristics, environmental constraints, and task behavior in the intended place of use. How should we design interactive systems? More specifically, how can we design the UI of these systems in a way that makes them both usable and useful [Erickson 2000]?

User-Centered Design (UCD) has been proposed in the literature to provide designers with a general approach for interactive system design, by making end-users and their experiences a focal point of the design process. Based on UCD principles, different design methods have been developed. These include Scenario-based design [Carroll 2000], Goal-directed design [Cooper 1999], Contextual design [Holtzbatt and Beyer 1998] and Participatory design [Ehn 1988]. These methods introduce techniques for evolving and documenting the design at various steps of the process. If a designer would like to model user experiences, tasks and the context of use – relevant techniques include personas, task analysis, scenarios, workflow modeling, and context analysis. Furthermore, if a designer would like to build a prototype, conceptual design, or detailed

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

design – relevant techniques include design guidelines, principles, style sheets, and patterns.

Although these methods share a common user-focused tenet, there exists a significant gap between current user analysis and modeling techniques, and the process of deriving a UI design. Ethnographic and empirical techniques are generally used to collect relevant user data to describe user experiences. These experiences are then captured in narrative form, but the derivation of a design from them is ambiguous and based on guided principles rather than a reproducible systematic method. Even if some techniques like storyboarding try to "walk" designers through relevant user tasks, they only address a subset of user experiences. There is little reproducibility of solutions and traceability to user experiences. Often, the final design is only the result of the designer's background and knowledge rather than the result of following a well-established and standardized method [Preece et al. 2002].

In [Seffah et al. 2005], the need to build a tighter fit between user experiences and design concepts is described as one of the main challenges in human-centered software engineering. To advance the state-of-the art, we require processes that are systematic, traceable and practical, but which leave room for design creativity. In this thesis we propose a UI design method and associated process to tackle this problem.

1.2. Research Statement and Objectives

Figure 1-1 illustrates the problem. User experience descriptions and UI conceptual designs are two major artifacts of UCD. User experiences encapsulate information about the user, their characteristics and interaction behavior; they are often captured in narrative form. Conceptual designs are early designs of the user interface, which only capture essential structural, behavioral and presentation-related details. In UCD, there is currently no way to systematically derive a conceptual design from user experiences. Typically, the design is reliant almost completely on the designer's intuition. This is especially problematic for novice designers who lack the background and training required to make trade-offs, judgments and interpretations towards a usable design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

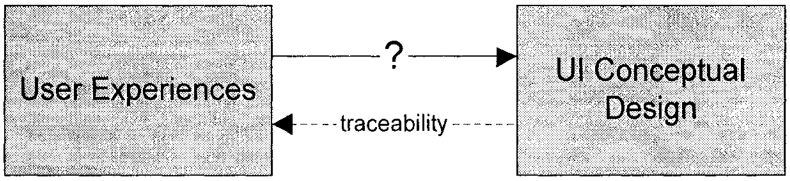


Figure 1-1. Current Problem of Deriving a Design from User Experiences

Narrowing the gap between user experiences and conceptual designs should be done in a stepwise fashion, through the use of domain-independent elements, and with tool support. We investigate personas and patterns as two complementary techniques which can be correlated for the purpose of narrowing this gap. Personas can be used to capture user experiences, while patterns can be used as "building blocks" to create conceptual designs. More precisely, our research is tailored towards the definition of a systematic process that derives a pattern-oriented design from persona descriptions, through a set of intermediate steps. By defining such a process and with tool support, it is possible to empower UI designers with concrete design solutions that can be traced back to the user experiences. This would be of great help in empirical or formal validations, as well as when design tradeoffs are to be made.

More specifically, the objectives of this thesis are:

(1) To define a systematic process to guide designers from user experiences to a UI conceptual design. Current interactive design methods based on the UCD philosophy provide a generic framework for UI design, including general suggestions, techniques and principles. They do not however, rigorously define processes and tools for translating inputs of the user analysis phase into concrete design solutions.

(2) To incorporate user needs, behaviors and experiences, as a key input to the design process. Current UCD methods do emphasize user-data collection using techniques such as ethnography and field studies; however they do **not** explicitly define how user data should impact the design of the user interface. In particular, it is unclear how particular user attributes and behaviors propagate into the actual design step.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(3) To capture and represent both "user experiences" and a "UI conceptual design" in a rigorous manner, using a suitable combination of representational models and techniques. As will be discussed further, we will use personas and patterns as the foundation for these representations.

To this end, the research questions addressed in this thesis are as follows:

> *How can we systematically generate a conceptual design from the model we have of the user experiences?*
> *How much of this process can be automated or computer-supported?*
> *What are the major steps in this process?*

We state our research hypothesis as follows:

> *The proposed process will lead to designs that can be argued as more usable designs satisfying the collection of user experiences.*

In this case, the null hypothesis is:

> *There is no relationship between employment of the proposed process and more usable designs satisfying the collection of user experiences.*

Furthermore, we assume that a taxonomy and valid library of HCI patterns reflecting best practices exist, and issues related to them are not part of this research work.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.3. Terminology

We define a number of terms below which will be used throughout the thesis.

Design Philosophy, Method and Process

A design *philosophy*, such as UCD, is characterized by a set of principles that all together aim to achieve a design goal, such as designing for usability. A design *method* is a set of techniques, processes, artifacts and tools that aim to support a design philosophy. A *process* is an organized set of activities, which transforms inputs to outputs. While a method is more generic, a process defines precise steps, allowing it to be easily applied with little need for interpretation.

UI Conceptual Design

A UI conceptual design at a high level of granularity can be viewed as a combination of structure, behavior, and presentation models of the user interface. It can also be viewed as an outline of the UI, which will further be developed during the detailed design stage. As an analogy, if one were to write a book, line spacing and typeface should not be of initial concern. Rather, the focus should be on content and structure. In essence, the table of contents represents the conceptual design of the book [Potosnak 1999; Baxley 2003].

User Experiences

User experiences is an umbrella term referring to a collection of information that covers a user behavior (observed when the user is in action), expectations, and perceptions – influenced by user characteristics and application characteristics. User characteristics include knowledge, experience, personality and demographics. Application characteristics include domain, content, language, visual design and interaction type.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Personas

Personas are descriptive models of the user, encompassing information such as user characteristics, goals and needs. They are captured in narrative form, and currently, there exists only general guidelines on how they should be represented. Personas are used primarily as a communication tool, with the hope that the information personas contain will "inspire" members of the development team to design interactive systems accordingly.

Design Patterns

Patterns are used in HCI and other design disciplines to capture essential details of design knowledge. The presented information is organized within a set of pre-defined attributes, allowing designers, for example, to search rapidly through different design solutions while assessing the relevance of each pattern to their design. Every pattern has three necessary elements, usually presented as separate attributes, which are a context, a problem, and a solution. Other attributes that may be included are design rationale, specific examples, and related patterns.

1.4. Organization of the Thesis

The organization of the remainder of this thesis is as follows:

In **Chapter 2** we discuss background and related work. In particular, we review User-Centered Design (UCD) and interactive design methods, current techniques for modeling users and their experiences, current techniques for building conceptual designs, and work done in narrowing the gap between user experiences and conceptual design.

In **Chapter 3** we present the results of our first empirical study, with 39 end-users. We propose a framework employing personas and patterns as the primary design directives in moving from user experiences to a conceptual design. We apply this framework to the redesign of a Bioinformatics website and compare the new design to the old design by performing usability tests with a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

prototype. The goals of the study were to evaluate and understand the steps involved in designing based on this framework, and to determine whether the development of a rigorously-defined process is substantiated.

In **Chapter 4** we propose a method for the design of interactive systems, based on our study results from Chapter 3. This method defines the three phases involved in moving from user experiences (denoted as UX) to a conceptual design: Persona Creation, Pattern Selection and Pattern Composition. In addition, models and artifacts (besides personas and patterns) which provide supplementary design input are discussed.

In **Chapter 5** we discuss the representational and knowledge requirements for the *UX-P Process*. First, we present the various elicitation techniques used to gather and represent these requirements. We performed interviews and a focus group with HCI experts. Second, we propose a persona model, pattern model, heuristics for creating personas, and criteria for selecting patterns.

In **Chapter 6** we define the UX-P Process as a way of refining the generic method proposed earlier. We overview the process, the inputs, and characterize each step of the three phases. Furthermore, we provide supporting examples for two different design contexts: Persona creation for a mobile game, and pattern selection and composition for a simple access website (targeted for users with low literacy level).

In **Chapter 7** we first present the *P2P (Persona to Pattern) Mapper*, a supporting environment for designers using our process. The P2P Mapper provides tools which automate various steps of the process and support designers in carrying out these steps. We then present the results of our second empirical study, with 22 end-users. We validate our work by designing a Bioinformatics application for 3D visualization of macromolecules. We follow each step of the UX-P Process, using our supporting environment. Based on the conceptual design, we build a prototype which is then evaluated with usability testing. The major goal of the study was to determine whether our hypothesis holds true.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Finally, **Chapter 8** summarizes our work, the major contributions, and presents future avenues for research in this area.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 2

Background and Related Work

In this chapter we provide a brief synopsis of user-centered design and other interactive design methods, with an emphasis on the existing "gap" between the two activities of gathering relevant user information and producing design solutions. We then overview related work for each of these two UCD activities – first, the existing "lightweight" techniques for capturing and representing user information and experiences, and secondly, the techniques for building conceptual designs. In this discussion, we focus on personas and patterns since they will be used as the main components of our proposed UI design method. Finally, based on varying perspectives, we highlight some research work done by others that attempt to provide a methodical link between user experiences and design.

2.1. UCD and Interactive Design Methods

User-Centered Design (UCD) is a philosophy surrounding the interactive design of usable systems, with the purpose of achieving product usability. The usability of a product is defined in ISO 9241, part 11 [1991] as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction[1] in a specified context of use." This definition relates to the quality of the interaction between the person who uses the product to achieve actual work and the product or software application itself, including its user interface. According to Mao et al. [2001, 2005], UCD advocates a multidisciplinary design approach built primarily on user involvement, helps with a better understanding of user and task requirements, and is based on iterative design and evaluation. UCD consists of modular processes, and is internationally recognized by ISO standards [ISO 13407 1998; ISO TR 18529

[1] (1) Effectiveness, defined as the accuracy and completeness with which specified users can accomplish specified goals in a particular context of use; (2) Efficiency, described as the resources expended to achieve accuracy and completeness of user goals; and (3) Satisfaction, expressed as how users and stakeholders perceive the comfort and acceptability afforded by the system in question [ISO 9241 1991].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2000]. A more detailed explanation of each activity relevant to our research is as follows:

- **Understand and specify the context of use.** In this stage, the following information is gathered, analyzed and documented: (1) The characteristics of the intended users; (2) The tasks the users will perform; (3) A hierarchical breakdown of the global task; (4) The overall goals of use of the system for each category of user; (5) The characteristics of tasks that may influence usability in typical scenarios, such as frequency and duration of performance; and (6) The environment in which the users will use the system.

- **Specify the user and organizational requirements.** In most software development lifecycle models, a major activity is to specify the functional requirements for the system. For user-centered design, it is essential to extend this activity to create an explicit specification of user and organizational requirements, in relation to the context of use description. This specification should be described with the following considerations: (1) The quality of the UI and workstation design; (2) The quality and content of the tasks of the identified users. This includes the allocation of tasks between different categories of users, as well as user comfort, safety, health and especially motivation; (3) Effective task performance especially in terms of the transparency of the application to the user; and (4) Required performance of the new system in relation to operational and financial objectives.

- **Produce design solutions.** The next stage is to create potential design solutions. This stage has the following objectives: (1) Using existing knowledge, such as standards, guidelines, and patterns, to develop a proposed design solution; (2) Making the design solution more concrete (using simulations, paper prototypes, mock-ups etc.); (3) Showing the prototypes to users and observing them as they perform specified tasks; (4) Based on user feedback improve the design; and (5) Iterate this process until the design objectives are met.

The above descriptions highlight the steps for gathering user information and producing design solutions as integral parts of UCD. However, it is unclear how designers and developers are expected to use the information gathered and apply it to the actual UI design. Evidently, there is a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

need to develop a workable solution to remedy the lack of a systematic and practical method for transforming such user information into concrete design solutions.

Based on the UCD philosophy, interactive design methods which advocate usability and are in current use within the HCI community, are: Scenario-based design, Usage-centered design, Contextual design, and Participatory design. However, none of these methods effectively address the process of deriving design solutions from user information (see Table 2-1). At present, this process is a complete human activity, with no automation whatsoever and lack of tool support. Appendix A describes each interactive design method in more detail.

Method	Description	Limitation
User-Centered Design (philosophy)	Multidisciplinary design approach built on user involvement, understanding user and task requirements, iterative design and evaluation. Has modular processes and follows ISO 13407 and ISO TR 18529 standards.	Unclear how information gathered as part of user requirements and context of use practices are applied to UI design of their systems.
Scenario-Based Design	Makes system use explicit by capturing one particular instance of user-system behavior. Scenarios depict stories about users, behaviors and actions, in a particular context.	Still a long way to go from stories captured as text to concrete design solutions. The resulting design cannot be mapped directly from scenarios.
Usage-Centered Design	Focuses on *usage* and employs models (user, task and content) at different levels of abstraction to drive design, with selective user involvement.	Abstract prototypes from the content model should be based on information contained within the user and task models. However, it is ambiguous how this progression occurs and how designs are actually derived.
Contextual Design	Concentrates on integrating user data into design through activities such as user environment design; where work functions and objects are organized as *focus areas* (has purpose, function, other foci links, interaction objects, constraints).	Fuzzy link exists between captured data and resulting design solutions. Focus areas represent possible UI constructs, but it is not well-defined how one arrives at actual design solutions.
Participatory Design	Promotes interactive system design as a creative activity with users and designers. Participatory techniques include collaborative design, ethnography and participative analysis of usability data.	User experiences are captured as narratives and incorporated into scenarios and storyboards. Design solutions are constructed from user opinions and designer experiences, with little focus on reuse of best practices in design. A defined process is missing (although user involvement in design is a positive step).

Table 2-1: Interactive Design Methods and Limitations

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.2. Modeling Users and their Experiences

Several techniques have been proposed within the HCI community to understand users and to model their needs. Since the term "user experiences" is multi-faceted, different techniques have different strengths and weaknesses. In this section, we describe three of them: Cognitive styles, user profiles and personas. Our main focus will be personas.

2.2.1. Cognitive Styles

A cognitive style refers to an individual's tendency to behave in a certain manner. It is a personality dimension described as a functional variation when an individual carries out a task. Functional variations include perception, memorization, learning behavior, judgment, decision-making and problem solving. They are consistent across different situations since they deal with intrinsic user characteristics. Furthermore, these variations can help determine the most effective interface for particular user groups, especially during more formative stages of interaction design [Muylwijk et al. 1983; Dufresne and Turcotte 1997].

Examples of cognitive styles include field dependence, impulsive reflective style, categorization style, and analytic-global style. As an example, field dependence relates to organization of information according to contextual demands. An individual can either have a field-dependent or field-independent cognitive style. Field-dependence is characterized by the difficulty in structuring ambiguous information and in restructuring new information (the latter is usually done be creating links with prior knowledge). Field-independence can be thought of as the opposite; the ability to readily structure new information even if it is ambiguous, and incorporating new information within the context of prior knowledge. To test for this style, users are given several exercises and asked if they remember shapes or other types of information when presented in both significant and insignificant contexts.

Modeling cognitive behavior using styles is a difficult task for HCI practitioners since it requires

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

some expertise in cognitive psychology and neuro-psychology. Finding models that are simple and applicable is not easy, and ironically, little work has been done to make them usable for novice designers [Byrne 2003]. Detailed testing with subjects is required, and this can be both time-consuming and impractical for designers. Most importantly, cognitive styles by themselves do not contain enough information to model users and their needs, from a UI design perspective. There are many other user characteristics such as user preferences, motivation, and even physical attributes that need to be considered during UI design which are not covered in this case.

2.2.2. User Profiles

User profiles model individual users. Factors which are taken into account are user characteristics, motivation and attitude, as well as context of use information. User characteristics consist of attributes such as color-blindness, disabilities, age and gender. Motivation and attitude include user opinions about computer use and technology adoption, willingness to learn, fear about new ways of doing things, and any other relevant information. Context of use information describes environmental characteristics (both the physical and cultural environment) in relation to the user and the system. Examples of the physical environment include type of region (urban/rural), mobility, lighting and workspace layout. Examples of the cultural environment include user language, literacy, and societal beliefs. In addition, corporate policies or customs are important cultural considerations of an organization.

To illustrate why such information is critical for UI design, we can take the example of a public health information site. If user profiles indicate that a high number of senior citizens use the site, then support for larger fonts may be a design decision worth exploring. In addition, if they are discretionary users who fear technology adoption, they will require "first-time ease-of-use" as part of the design objective otherwise end-users will simply not use the system [Mayhew 1999]. Finally, if they have slower internet speeds, the context of use consideration may dictate the use of simpler computer graphics requiring less time to download.

User profiles contain a great deal of information. However, there is no widely-accepted

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

representation and standard. They are described in statistical and narrative form. It is difficult for designers, especially for inexperienced designers, to know what information is useful and how it should affect design decisions. Furthermore, user profiles model individual users and not groups of users. Statistical analyses are then used to find the "average" user profile. This is very similar to market segmentation in the field of marketing, where the purpose is to overcome consumer heterogeneity. By trying to design for such "average" users (usually found in the statistical center of identified market segments), crucial user information may be lost. User profiling using market segmentation and usability testing is an invaluable tool for identifying groups of people – for example, identifying those who are most likely to use a certain website. It may even answer why they will use the website. However, it will not provide enough insight into how the website needs to work and how it may be best designed.

2.2.3. Personas

Alan Cooper, the father of Visual Basic, proposed the use of personas in software design. His original work in 1999 [Cooper 1999] brought the concept of personas from marketing to UCD; so as to redirect the focus of the development process towards end users and their needs. His work emphasizes personas as being fictitious characters, based on composite archetypes, and encapsulating "behavioral data" gathered from ethnography and empirical analysis of actual users. Archetypes have been used in marketing research both as an alternative and as an extension of traditional market segmentation and user profiling. Instead of modeling only "average" users, personas also take into account boundary cases. The underlying belief is that all consumers are a mixture of certain types of users.

Each persona should have a name, an occupation and personal characteristics such as likes, dislikes, needs and desires. In addition, each persona should outline specific goals related to the project. These goals can be personal (e.g. having fun), work-related (e.g. hiring staff), or practical (e.g. avoiding meetings) [Tahir 1997]. Personas are intended to help developers better understand both the users and context of use for a planned tool or interactive system. Cooper [1999] argues that designing for any one external person is better than trying to design vaguely for everyone. He

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

also believes that for each project, a different set of personas should be constructed. This is because each project targets different users in different contexts of use.

Personas can be used by different team members during various stages of software development. Cooper describes a 7-step "standardized" process to constructing personas, from which identified goals can be translated into design solutions. Personas are derived from the user's interaction behavior, described in the requirements gathering stage. Missing data are filled by using other sources of information, gathered from stakeholders. The process is as follows: (1) Re-examine the persona hypothesis based on real-world observation and results, (2) Map interview subjects to "behavioral variables", (3) Identify significant "behavior patterns", (4) Synthesize characteristics and relevant goals, (5) Check for completeness, (6) Develop narratives, and finally (7) Designate persona types.

The Persona Hypothesis is the first definition of user types and profiles, based on the particular product and domain under consideration. It becomes the basis for the initial set of interviews, with experts concentrating on questions addressing three main topics: The different types of users; variations in their needs and behaviors; and the behavior and environment ranges to be explored. The results of data collection are then compared to assumptions made in the persona hypothesis. After a complete re-examination of the hypothesis, a set of relevant behavioral variables is listed, along with demographic variables. These behavioral variables refer to distinguishing elements of interaction behavior. For example, in the case of an e-commerce application, users are "service-oriented", "price-oriented", or fall somewhere in between.

Each user should then be mapped to the complete set of behavioral variables and their associated ranges. The precision of this mapping is not as critical as the actual placement of users, since the latter is important for grouping purposes. Behavioral patterns are a result of the grouping of particular subjects occurring across multiple ranges or variables. A significant behavioral pattern emerges from a set of subjects grouped within 6-8 different variables, and forms the basis of a persona. It is important to note that some specialized roles may exhibit only one specific pattern. In addition, there must be a logical or causative connection between the grouped behaviors for a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

pattern to be considered valid, and not just a spurious correlation.

The next phase entails synthesizing details obtained from data, namely the characteristics and goals, with the persona descriptions. For each observed pattern of behavior, details based on the ethnographic studies are added. Examples include descriptions of the potential usage environment, a typical workday, current solutions and frustrations, interrelationships with others, and most importantly, user goals. At this point, it is cautioned that although personas are viewed as fictional characters, excessive irrelevant personal detail should be avoided in the descriptions.

To check for completeness in the constructed personas, we identify gaps in the descriptions. If certain behaviors seem to be missing, additional ethnographic research may be necessary. In addition, Cooper suggests using a checklist which can ensure maintainability of a manageable set of personas. First, it is important to add political personas for stakeholders. Second, if two personas only vary by demographics, we should eliminate one of them or make them more distinct. Finally, we should check that each persona varies from all others in at least one significant behavior. Up to this point, one might have used bullet points for depicting personas. Cooper argues that if personas are to have an impact on designers and developers, they should depict "real" people. Therefore, we should develop third-person narratives, as 1-2 pages of prose, which are a better communication tool than simple bullet points.

The first and most obvious strength in Cooper's work is that he defines a basic framework and general steps for persona creation, which brings some kind of organization and format to capturing user characteristics and experiences. Behavioral variables, behavioral patterns, and groupings are important persona constituents which have been identified, in spite of the fact that they are not clearly defined. His focus on interaction behavior associates personas to a specific context of use. Furthermore, he claims that the identified goals during his process can be translated into design solutions. Secondly, he links personas albeit loosely to what he calls "research". This is the associated data collection as captured by different methods such as ethnography. Thirdly, his focus on the customer, including his primary focus on market segmentation to initially help with the "persona hypothesis" is a practical approach for industry.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Assuming that this kind of market data already exists for a specific project, one can easily proceed to the necessary task of persona development, increasing its attractiveness for industry applications. Finally, his overall method follows the principles of goal-directed design (discussed in section 2.5.1), making it an intrinsically user-centered design technique, with a strong focus on user goals.

However, saying that, there are a number of inherent weaknesses in Cooper's work on persona. First, the persona descriptions are expressed in narrative form which is inherently imprecise, subjective and story-like. As a result, they are open to interpretation by the designer and do not facilitate any form of quantifiable analysis. In essence, the method does not lend itself to extracting any useful information for the novice designer. It is not clear as to how data translates to actual behavioral variables and patterns. Although Cooper encourages the use of goal-directed design, he offers no concrete method to transform the actual goals to design solutions resulting in the designer having to resort to using personal intuition and experience.

Secondly, the method lacks both best practice templates and lists of possible behavioral identifiers. The grouping of behavioral patterns (comprising many different identified behavioral variables) is subject to imprecise and subjective interpretation which runs the risk of improperly identifying causalities between behaviors. While some may be obvious, especially in well researched domains, others are not. In addition, one must be vigilant against defining false patterns. For example, in the field of Bioinformatics, there is a logical connection between biologists working in drug discovery and time spent analyzing 3D molecular structural data. However, there may not be any connection if biologists who are working in drug discovery all happen to spend an unusual amount of time checking their e-mail.

Thirdly, Cooper states that one should strive to map a complete set of variables. In practice, this is very difficult as there may be simply far too many variables to consider. In addition, it is difficult to validate whether or not the set of variables is in fact complete. Furthermore, the use of demographic variables and user-cluster weightings in persona development is underplayed. In fact, Cooper states that grouping is important only in relative position to the others in the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

behavioral range. This ignores the quantifiable position of individual users resulting in limited precision in the definition of user characteristics and experiences.

Pruitt and Grudin [2003] describe two long-term studies which they carried out at Microsoft Research using personas. The first study was on MSN Explorer and lasted 10 months. The second study was on MS Windows, on a much larger scale and has lasted over 2 years (it is still ongoing). From a development perspective, personas were used to engage team members, as an effective communication tool, and to convey a broad range of user data in a simple form. In addition, personas were attractive for the study since they focus on both design and use aspects whereas most other methods currently employed in industry do not. However, the studies highlighted several problems with personas: (1) Persona characters are not always "believable" since they are not based on thorough data analysis, (2) They are not communicated well visually or narratively, and (3) It is unclear how they can be used.

As suggestions to remedy some of the shortcomings with current use of personas, Pruitt and Grudin extend Cooper's technique by complementing it with other usability and data collection methods. Cooper's personas focus only on an initial investigation phase for data – which includes interviews and ethnography to create more detailed characters at the beginning of persona creation. Ongoing data collection is actually not encouraged since it is "time-consuming and expensive" [Cooper 1999]. In contrast, Pruitt and Grudin promote the ongoing use of qualitative and quantitative data to select and evolve personas during the entire design lifecycle. To this end, their approach consists of:

- Large-sample market segmentation studies to collect data. The highest priority segments are filtered with user research techniques such as field studies and interviews. If available, information on the international market, accessibility data and anti-personas are incorporated.
- "Affinity" sessions with team members to physically cut data points and form groups of related findings across studies. In addition, "sanity checks" can be used with real users who match personas on high-level characteristics to see how well they correspond with low-level characteristics. Since multiple data sources are used for persona creation, some may not be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

justified or compatible.

- A *foundation document* for each persona with the purpose of making the links between persona characteristics and data more explicit. This is an overview of the persona, with references to other documents that contain additional information. As an example, references can be found to scenarios found in the feature specifications. In addition, persona stories include qualitative data and observational anecdotes (as opposed to simply fictional information).
- The persona as part of the development process and not just as a discussion tool. A weighted feature-persona priority matrix can help prioritize features as part of the product development lifecycle. In addition, the Microsoft team developed some basic tools for internal use – spreadsheet tools and document templates – to support persona creation. Visual illustration of a persona based on a representative user helps to intensify communication about users with the entire development team.

In summary, Pruitt and Grudin's approach encourages a more "global" use of personas. This includes attempts to integrate personas in the software development process and by establishing relationships with other data sets through the use of artifacts such as feature-persona matrices, foundation documents, and task descriptions (although the latter is mentioned, specific examples are not provided). In addition, a focus on ongoing qualitative and quantitative analysis is a central theme of their work. However, there is little discussion on what kind of detailed information is contained in their personas, how they are represented, and how they are mapped to actual data sets. Furthermore, it is unclear if and how precise interaction behavior is addressed in their personas.

Based on the above work, Courage and Baxter [2005] define a set of persona components. These components are in text format, and can act as a guide in building personas. As will be discussed later, we fine-tuned these components to better fit the requirements of our process (see Table 2-2).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Persona Components	Description
Identity	Include a first and last name, age and other demographic information.
Status	Whether the user is a primary, secondary, tertiary, or anti-user of the application. Typically, only primary and in some cases, secondary users are included.
Goals	Besides goals related to the application, it includes personal and professional goals as well.
Knowledge and Experience	Knowledge and experience including education, training, and specialized skills. This should not be limited only to the application.
Tasks	Frequency, importance and duration of most important tasks related to the application.
Relationships	Include information about user associates, since this could give insight on other stakeholders.
Psychological profile and Needs	Include information about cognitive and learning styles, as well as needs such as guidance and validation of decisions.
Attitude and Motivation	Include information about the user's attitude to information technology and level of motivation to use the system.
Expectations	Information about how the user perceives the system works, and how the user organizes information related to his/her task, domain or job.
Disabilities	Any disabilities, such as color-blindness, related to mobility, eyesight (wears contacts), etc.
Photograph	Include a photograph which fits with the name.

Table 2-2: Persona Components (adapted from Courage and Baxter [2005])

In the next section we will overview related work in building conceptual designs.

2.3. Building Conceptual Designs

A UI conceptual design at a high level of granularity can be viewed as a combination of structure, behavior, and presentation models of the user interface (see Figure 2-1). The term "behavior" is often understood by writing informal descriptions. By using a model or a combination of models, we can formally capture the interactive behavior to be further refined according to our needs. As an example, we can use decision trees to describe the choices faced by a user during system interaction. The UI structure can also be modeled in various ways. Cognitive load on the end-user is often a consideration of which designers must be cognizant. For web applications, depth and breadth of "trees" depicting site organization can be used. Finally, the UI presentation includes descriptive information about elements such as layout, style and textual information. It is in part based on structural and behavioral considerations, as well as design constraints such as device limitations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

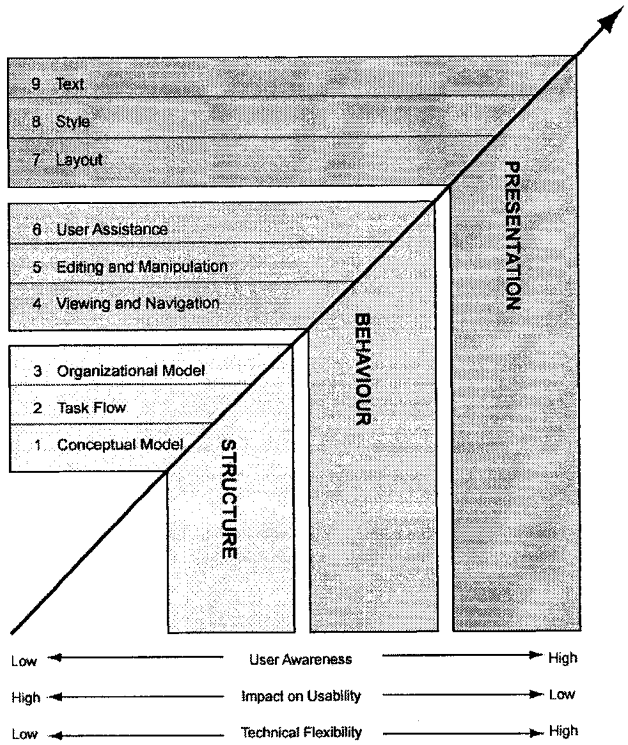


Figure 2-1: Constituents of a UI Conceptual Design (adapted from Baxley [2003])

The resulting design should be a close match to the user's mental model. An example of a basic conceptual design can be found in Figure 2-2.

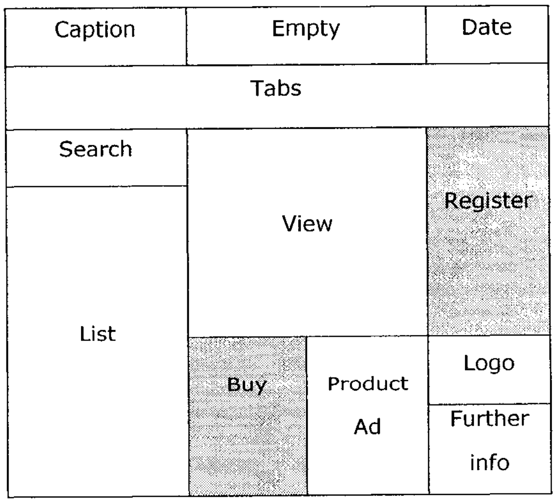
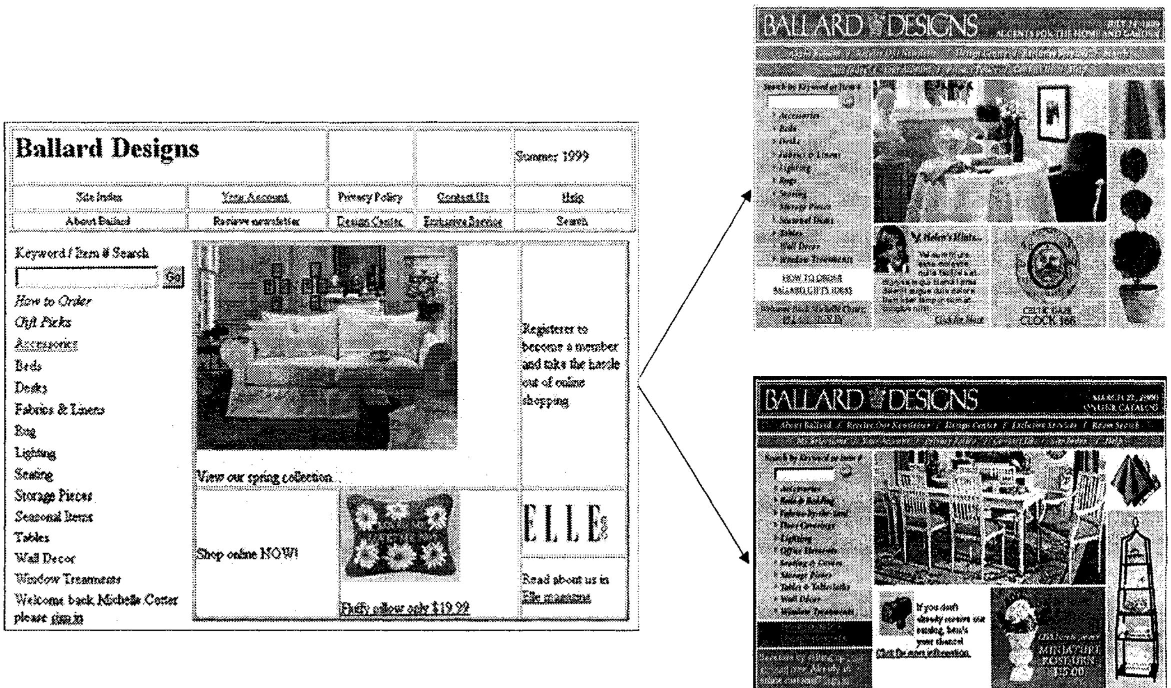


Figure 2-2: Basic Conceptual Design of a Website

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Figure 2-3a illustrates a more advanced conceptual design for the same website. Conceptual designs are then further refined and implemented into various detailed designs, as portrayed in Figure 2-3b.



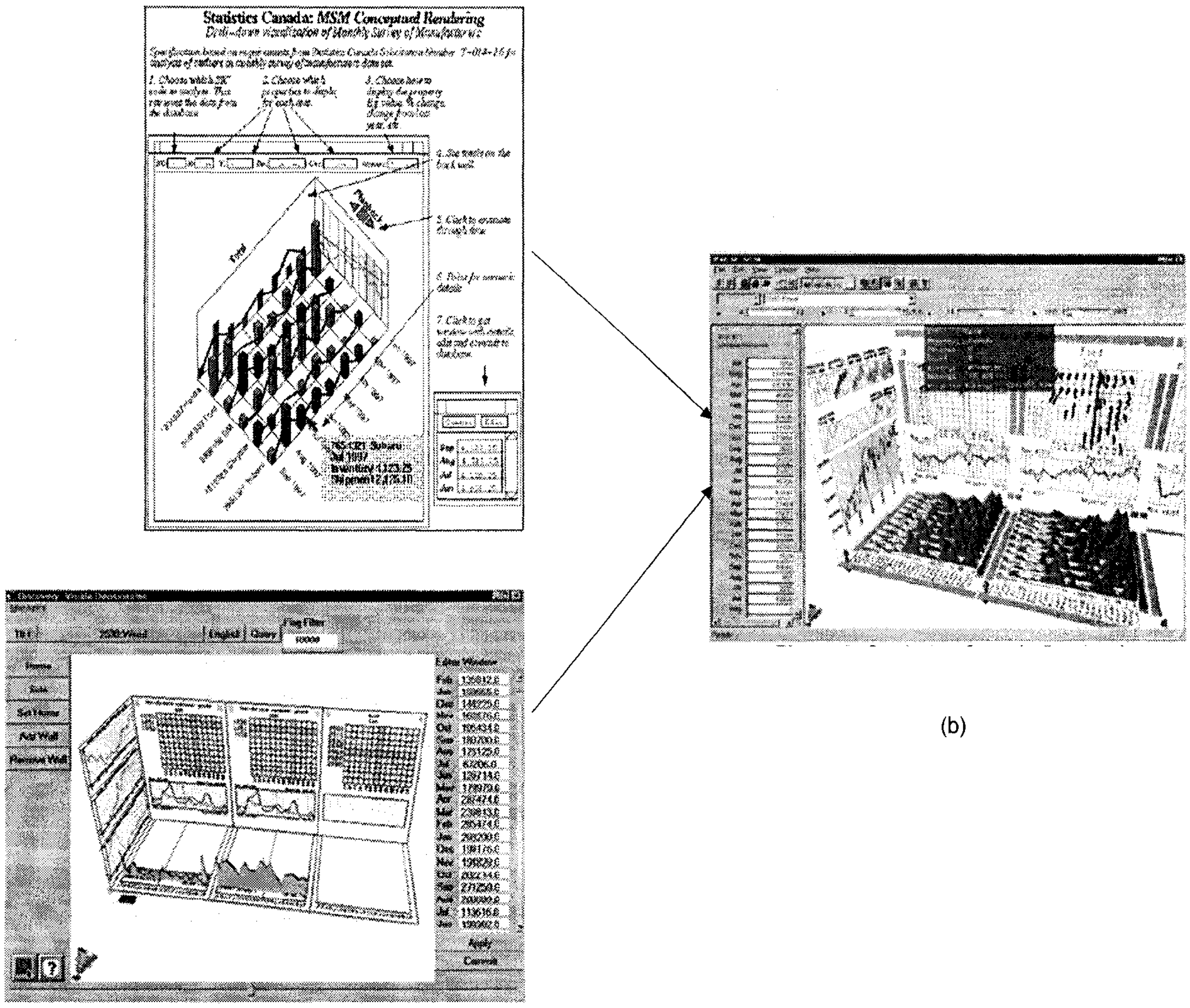
(a)

(b)

Figure 2-3: Conceptual and Detailed Designs of a Website [Najjar 2000]

Figure 2-4a portrays a more complex example. Two different conceptual designs for the same Statistics Canada visualization application are illustrated – the top one is paper-based, and the bottom one is a software prototype. Figure 2-4b is the detailed design, which could be the result of either conceptual design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



(b)

(a)

Figure 2-4: Conceptual and Detailed Designs of a Visualization Application

User interface design is a repetitive activity and there is ample scope for reuse of design knowledge. Techniques used to guide designers in building conceptual designs include design principles, guidelines, style sheets and patterns. Detailed designs are then created based on conceptual designs, as illustrated in the diagrams above. Creation of detailed designs will not be in the scope of this research work, but further information using patterns as a technique can be found in [Sinnig et al. 2005].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In past years, guidelines formed the main source of UI design information. Their limitations prompted designers to search for alternative resources and tools. As a result, both claims and patterns were subsequently introduced. In the next two sub-sections we briefly review guidelines and claims. Section 2.4 will be dedicated to a more thorough review of patterns. As will be discussed, patterns contain numerous advantages which make them particularly attractive for UI design.

2.3.1. Guidelines

Among UI designers, "design guidelines" [Apple 2003; Microsoft 2004] are used to disseminate usability knowledge and ensure a degree of consistency across applications and projects. These guidelines often take the form of style guides and are usually platform-specific. They describe how different kinds of windows should look and interact with the user for basic tasks such as choosing from lists or menu controls. Good examples are the *Macintosh Human Interface* Guidelines [Apple 2003] and the *Java Look and Feel* Design Guidelines [JAVA 2006]. An example of the latter is as follows:

Toolbar

> A **toolbar** is a collection of frequently used commands or options that appear as a row of toolbar buttons. Toolbars normally appear horizontally beneath a primary window's menu bar, but they can be dragged anywhere in the window or into their own window. Toolbars typically contain buttons, but you can provide other components (such as text fields and combo boxes) as well. Toolbar buttons can contain menu indicators, which denote the presence of a menu. Toolbars are provided as shortcuts to features available elsewhere in the application, often in the menus.

Guidelines have not realized their full potential and have had minimal impact on the design of user interface software [Gould et al. 1991; Souza and Bevan 1990]. Apart from not adequately addressing concerns facing designers, such as which guidelines should be used under what circumstances [Henninger et al. 1995], studies have shown that interface guidelines suffer from being too abstract to be applied directly [Tetzlaff and Schwartz 1991; Thovtrup and Nielsen 1991]. In addition, they are described in narrative text and focus on physical UI design elements, making them more like a general reference for design practices rather than a practical design tool.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

From a usability standpoint, guidelines provide general and unstructured information, and are not problem-based. They do not effectively promote reuse because they are too tailored to a particular toolkit or technology [Borchers 2001]. Most guidelines fall short of the goal of putting the accumulated knowledge of user-centered design at the fingertips of everyday designers, often becoming a static document read only by human factors specialists.

2.3.2. Claims

Introduced in the last decade, "claims" [Sutcliffe 2000] are another means to capture and disseminate HCI design knowledge. A claim is a unit of design knowledge that is associated with a specific artifact and usage context, providing design advice and possible trade-offs. Claims are powerful tools because, in addition to providing negative and positive design implications, they contain both theoretical and cognitive rationale. They also contain associated scenarios which provide designers with a concrete idea of the context of use. An example of such a claim for a safety-critical application is given in Table 2-3.

Claim: Safety-critical application
Claim ID: Rare event monitor Target artifact: User interface for a chemical analysis instrument control system
Description: Infrequent, dangerous events are detected by the system and a warning is issued to the user; in this case operational failures in a laser gas chromatograph control system.
Upside: Automatic detection of dangerous events relieves the user of constant monitoring; automatic detection and warning gives the user time to analyze the problem.
Downside: Issuing too many warnings may lead the user to ignore critical events; automated monitoring may lead to user overconfidence in the automated system and decrease their situation awareness.
Scenario: No events are detected in the laser emission controller or power supply, so the system gives an audio warning to the user and visually signals the location of the problem on a diagram of the instrument.

Table 2-3: Example of a Claim

Although claims contain theoretically sound HCI information and are rooted in cognitive rationale, they are tied to specific domains of use and are somewhat narrowly defined within

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

scenarios and examples. This leads to two problems with claim use. First, the creation of generic claims that will effectively support a reuse paradigm is a difficult feat. The specificity of claims allows them to contain useful information, but is somewhat of a hindrance in the world of design reuse. Secondly, applying current claims to desired application contexts requires careful matching. Finally, the lack of software tools that support the use of claims is a detriment to their use. For experienced designers, simply retrieving relevant information is problematic. For novice designers, the problem is even more difficult since they are often unsure of what kind of information is relevant in the first place.

2.4. Patterns and Pattern-Oriented Design

Patterns overcome some of the limitations associated with guidelines and claims. They are an effective technique for collecting design practices, and for using these practices in creating a conceptual design. In what follows, we will discuss different aspects related to patterns and pattern-oriented design.

2.4.1. Original Ideas about Patterns in Design

The building architect Christopher Alexander first introduced the concept of design patterns in the late 1970s. In his two books [Alexander et al. 1977, Alexander 1979], he discusses the capture and use of design knowledge in the format of patterns, and presents collections of pattern examples to help architects and engineers with the design of buildings, towns, and other urban entities. To illustrate, Alexander proposes an architectural pattern called *Wings of Light* [Alexander et al. 1977], where the problem is:

> "Modern buildings are often shaped with no concern for natural light - they depend almost entirely on artificial light. But, buildings which displace natural light as the major source of illumination are not fit places to spend the day."

Amongst other information such as design rationale, examples, and links to related patterns, the solution statement is:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

> "Arrange each building so that it breaks down into wings which correspond, approximately, to the most important natural social groups within the building. Make each wing long and as narrow as you can - never more than 25 feet wide."

Similarly to the above example, all of Alexander's patterns address recurrent problems that designers face by providing a possible solution within a specific context. They follow a similar structure, and the presented information is organized into pattern attributes, such as *Problem* and *Design Rationale*. Most noteworthy, the presented solution statement is abstract enough to capture only invariant properties of good design. The specific pattern implementation is dependent on the design details and the designer's creativity [Dix et al. 2003]. In the *Wings of Light* example, there is no mention of specific details such as the corresponding positions of wings to one another, or even the number of wings. These implementation details are left to the designer, allowing for different instances of the same pattern solution. In addition, Alexander recognized that the design and construction of buildings required all stakeholders to make use of a common language for facilitating the implementation of the project from its very beginnings to completion. If organized properly, patterns could achieve this for all the participants of a design project, acting as a communication tool for design.

The idea of using Alexandrian-type patterns as a design tool has been quite influential in a variety of domains in the last decade, including software engineering. His pattern framework has been applied extensively to object-oriented programming, and inspired a different way of thinking in which design knowledge is captured and reused effectively. Alexander's influence is apparent in Gamma et al.'s [1995] book, "Design Patterns: Elements of Reusable Object-Oriented Software". This book inspired the software engineering community to take a closer look at the concept of patterns as a problem-solving method for object-oriented design.

2.4.2. HCI Patterns

In HCI, patterns have also been introduced as a tool to capture and disseminate proven design knowledge, and to facilitate the design of more usable systems. Patterns aim to capture and communicate the best practices of user interface design with a focus on the user's experience and the context of use. As a result, they are an attractive UCD technique, with interesting

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ramifications for designing across a variety of contexts. Patterns are applicable to different levels of abstraction such as the user-task model or the concrete presentation of the user interface. They are a great source of interest not necessarily because they provide novel ideas to the software engineering community, but because of the way that they package already-available design knowledge. This way of presenting information to designers and developers allows the reuse of best practices, and avoids reinventing the wheel each time.

Patterns only capture essential details of design knowledge, and abstract away from superfluous and platform-specific design information. In addition, the presented information is organized within a set of pre-defined attributes. Every pattern has three necessary elements, which are: A context, a problem, and a solution. The context describes a recurring set of situations in which the pattern can be applied. The problem refers to a set of forces, i.e., goals and constraints, which occur in the context. The solution refers to a design form or a design rule that can be applied to resolve the problem. The solution describes the elements that constitute a pattern, the relationships between these elements, as well as their responsibilities and collaboration. Other attributes that may be included are additional design rationale, specific examples, and related patterns.

Patterns alleviate many of the shortcomings associated with guidelines. Above all, they are a good alternative to guidelines because they are problem-oriented, but not toolkit-specific. They are more concrete and easier to use for novice designers. Guidelines can be vague, whereas patterns are more structured and the knowledge is placed in a context. The designer is told when, how and why the solution can be applied. Since patterns are context-oriented, the solution is related to a specific activity.

Patterns have a more complementary association with claims; this in contrast to their somewhat antagonistic relationship with guidelines. Claims are tightly bound to specific domains of use, but contain valuable information including design trade-offs, and a possibility is to use them to complement patterns creating a "package of reusable knowledge" [Sutcliffe 2000]. Such detailed

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

information can be incorporated when the pattern is instantiated to a specific context of use. Furthermore, details from claims about design and cognitive rationale, including scenario descriptions, can provide additional information to designers when combining patterns to create comprehensive designs.

Overall, patterns have a number of benefits, including [Dix et al 2003; Taleb et. al. 2006]:

- They are a relatively intuitive means to document design knowledge and best practices;
- They are straightforward and readable for designers, developers and other stakeholders, and can therefore be used for communication purposes;
- They come from experiments on good know-how and were not created artificially;
- They represent design knowledge from different views, including social and organizational aspects, conceptual and detailed design;
- They capture essential principles of good design by telling the designer what to do and why, but are generic enough to allow for different implementations.

This last property is an especially discriminating characteristic of patterns, allowing them to give rise to different implementations of the same design solution. Different implementations are necessary to support variations in design look and feel, platform preference and usage context. An example is *Overview and Detail* (see Table 2-4), a pattern for visualization environments. This pattern can be implemented differently by the designer, depending on variations in data and usage context. To illustrate, *Windows Explorer* and *Google Maps* have two different implementations. In Windows Explorer, the user is provided with two views – one which presents a hierarchical overview of folders, and the other, the contents of the selected folder. In Google Maps, the user is also provided with two views of the data – an orienting view of the selected area presented as a corner map, and a detailed view of the same geographic location.

In addition to the benefits described above, two cardinal properties of patterns have made their use increasingly valuable for designers. First, patterns include user-centered values within their

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

rationale. Second, the concept of patterns and their associated pattern languages are generative, and can therefore support the development of complete designs [Dix et. al. 2003]. The remainder of this section will look at how these two properties have allowed patterns to evolve from a simple compilation of "best practices" to a powerful tool for designers, to be used as building blocks for creating a conceptual design.

Title	**Overview and Detail**
Context	The dataset is large, too large for all the details to fit in a single view, and there is a need to view details about subsets of data items. The data can be viewed at one or more levels of abstraction e.g. directories and files within a directory, aggregated document content and detailed document content, etc. Alternatively the dataset may be large and continuous but only a subset can be viewed at any one time e.g. map data.
Problem	How to display the entire contents of a large dataset at once, allow users to explore the dataset, and at the same time show details about subsets of items.
Solution	**Show an overview of the entire dataset together with some visual indication as to which part of the dataset is currently being viewed. Show details about subsets of items in a separate view.** The overview can be a scaled version of the main view, i.e. a spatial zoom, or some other representation, i.e. a semantic zoom. Since the overview tends to display a higher number of data items than any more detailed view it is necessary to use simple glyphs that minimize clutter, maximize use of screen space and portrait the data attributes most relevant to the task.
Examples	Windows ExplorerTM Google Maps
Other Attributes	Forces, Related Patterns, Design Rationale

Table 2-4: HCI Pattern for Visualization Environments

2.4.3. HCI Pattern Languages

In the literature, the term pattern language is used to refer to an organized collection of interrelated patterns. Just as words must have grammatical and semantic relationships to each other in order to create sentences with meaning, design patterns must be related to each other in order to form meaningful design constructs. Pattern languages are a structured method of describing good design practices and are a means to traverse common HCI problems in a logical way, describing the key characteristics of effective solutions for meeting various design goals. Furthermore, they act as a communicative design tool and give rise to many different paths

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

through the design activity.

A number of pattern languages have been suggested in HCI. For example, van Duyne's "The Design of Sites" [van Duyne 2003], Welie's Interaction Design Patterns [Welie 2003], and Tidwell's UI Patterns and Techniques [Tidwell 2003] play an important role and wield significant influence in the HCI community. In addition, specific languages such as Laakso's User Interface Design Patterns [Laakso 2003] and the UPADE [2] Web Language (see Appendix B) have been proposed. Different pattern languages have been published [Engelberg and Seffah 2002; Wilkens 2003; Welie 2003], including patterns for web and mobile design, for navigation in large information architectures as well as for visualizing and presenting information.

We have proposed [Javahery 2003] that pattern languages should have three essential elements. First, the language should contain a standard pattern definition. One format for defining patterns was presented in Table 2-4 – with the common attributes Context, Problem, Solution, Forces, Related Patterns, and Examples. Secondly, the language should logically group patterns. Tidwell [Tidwell 2002] organizes her patterns according to different facets of UI design; categories include Content Organization, Navigation, Page Layout, and Actions/Commands. Another example is the Experiences pattern language (Figure 2-5), developed by Coram and Lee [1998], which concentrates on the user's experience within software systems. The main focus is on the interactions between the user and the interfaces of software applications. Patterns are grouped according to different focus areas and user interface paths such as interaction style, explorable interface, and symbols. Thirdly, pattern interrelationships should be described. In Experiences, the relationships between the patterns are mapped and indicated by arrows, creating a sort of "flow" within the language. This is illustrated in Figure 2-5.

---

[2] Usability Pattern Assisted Design Environment (UPADE) is a web language and design environment developed by Concordia University's Human-Centered Software Engineering Group

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

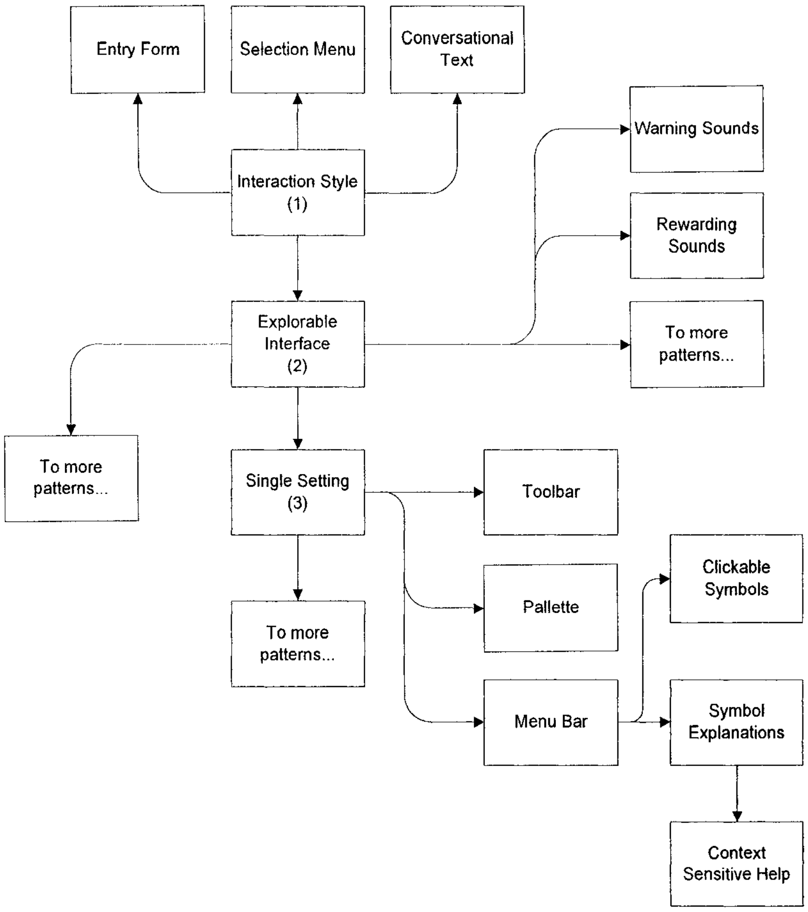


Figure 2-5: The Experiences Pattern Language

Distinguishing between different types of relationships reinforces the generative nature of pattern languages, and supports the idea of using patterns to develop complete designs. However, for designers to be able to use patterns effectively and with efficacy to solve problems in HCI and interactive system design, patterns need to be intimately related to a design process. Based on the design problem, pattern languages should provide starting points for the designer, and a means to systematically walk the designer from pattern to pattern.

For example, in Experiences, the meta-pattern *Interaction style* (denoted with "(1)" in Figure 2-5) is the first pattern that leads the designer along the major paths through the language. The design advice [Coram and Lee 1998] for this pattern includes studying the user and environment, working with the user to determine what interaction style is best, and keeping the interface simple and consistent. This pattern is connected to four other patterns as indicated by arrows (Entry

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Form, Selection Menu, Conversational Text, and *Explorable Interface*). Based on the context of use, the designer is free to choose any of these patterns to incorporate into the design. This is a repetitive process as some patterns, such as *Explorable Interface*, are subsequently connected to even more suggested patterns.

Although the Experiences Language showed the beginnings of associating its patterns to a design process, it was regrettably not developed in its entirety. In the next section, we will present some attempts at further linking pattern languages to a UI design process.

2.4.4. Patterns and the User Interface Design Process

The interface design of an interactive system can be a challenging task – and especially so when a project involves different design participants and stakeholders. Successful designs require individuals to communicate their concepts and ideas, building a common forum for the discussion of already-available design practices. As in any culture or society, the HCI community needs a common ground for such communication and dissemination of knowledge. Designers focus on the creation of an artifact that integrates various behavioral theories and technologies. This is done without regard to the evaluation of individual variables that may affect the design [Zimmerman et al. 2004]. Usability experts take a more scientific approach, looking at specific behavioral and design elements that best satisfy the requirements. Software developers are interested in finding an applicable design and implementing it correctly in the most efficient manner, and are often not familiar with usability engineering techniques and human interaction theories [Myers and Rosson 1992].

This is a proving ground for patterns as they provide a mechanism to successfully integrate and satisfy the different goals of all individuals involved in the design process, crossing cultural and professional barriers, and overcoming limitations in communication. Patterns are presented consistently, are easy to read, and provide background reasoning. They act as a *lingua franca* [Erickson 2000] for design, which can be read and understood by all. Thomas Erickson [2000] discusses the potential of this as a way of making communication in design a more "egalitarian

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

process", with the focus relying less on technical design issues, and more upon broader design problems and solutions. A lingua franca facilitates discussion, presentation, and negotiation for the many different individuals who play a role in designing interactive systems.

Acting as a communicative vehicle, pattern languages are an interesting tool which can guide software designers through the design process. However, there exists no commonly agreed upon UI design process that employs pattern languages as first class tools. Several people have tried to link patterns to a process or framework, bringing some order to pattern languages, and suggesting that potentially applicable patterns be identified early on based on user, task and context requirements. A pattern-driven design process should lead designers to relevant patterns based on the problem at hand, demonstrate how they can be used, as well as illustrate combinations with related patterns.

We will review three user-centered design approaches that are driven by patterns. First, in the Pattern-Supported Approach (PSA) Framework (Figure 2-6), HCI patterns are used at various levels to solve problems relating to business domains and processes, tasks, structure and navigation, and GUI design [Lafrenière and Granlund 1999]. The main idea that can be drawn from PSA is that HCI patterns can be documented, identified and instantiated according to different parts the design process – assisting designers as early on as during system definition. For example, during system definition or task and user analysis, depending on the context of use, we can decide which HCI patterns are appropriate for the design phase. Although PSA shows the beginnings of associating patterns to the design process, pattern relationships and their possible impact on the final design are not tackled.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

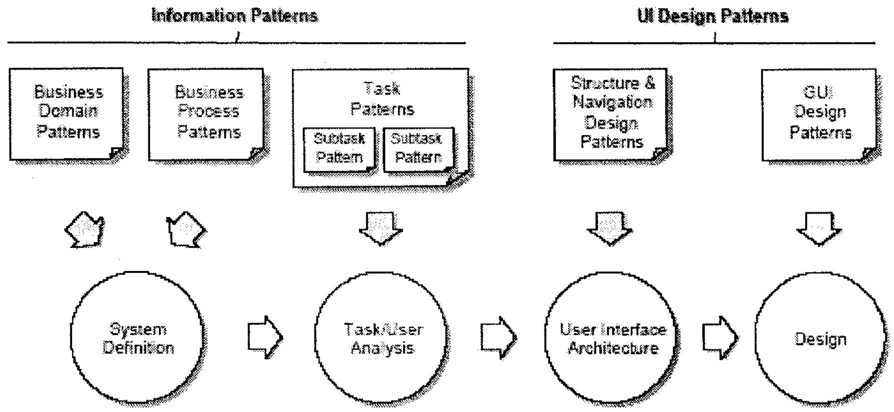


Figure 2-6: The Pattern-Supported Approach Framework

van Duyne et al. [van Duyne 2003] describe a second approach, where patterns are arranged into 12 groups that are available at different levels of web design. Their pattern language has 90 patterns that address various aspects of web design, ranging from creating a navigation structure to designing effective page layouts.

Step	Pattern Groups	Description	Pattern Examples
A	Site Genres	Construct particular site type	Personal e-commerce Nonprofits as networks of help
B	Creating a Navigation Framework	Choose patterns to navigate, browse and search on the site	Multiple ways to navigate Task-based organization
C	Creating a Powerful Homepage	Design the homepage based on user needs	Homepage portal Up-front value proposition
D	Writing and Managing Content	Manage content and address user accessibility	Page templates Internationalized and local content
E	Building Trust and Credibility	Address issues dealing with trust and credibility	Site branding Fair information practices
F	Basic E-Commerce	Create a good customer experience for e-commerce	Quick-flow checkout Clean product details
G	Advanced E-Commerce	Incorporate advanced e-commerce features	Featured products Cross-selling and up-selling
H	Helping Customers Complete Tasks	Structure your site to improve task completion	Process funnel Persistent customer sessions
I	Designing Effective Page Layouts	Create clear, predictable and understandable layouts	Grid layout Expanding-width screen size
J	Making Site Search Fast and Relevant	Design interaction so that user searches are effective	Search action module Straightforward search forms
K	Making Navigation Easy	Display helpful navigation elements	Unified browsing hierarchy Action buttons
L	Speeding Up Your Site	Incorporate patterns to make your site look and feel fast	Low number of files Fast downloading images

Table 2-5: Pattern Groups Ordered According to a Web Design Process [van Duyne 2003]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The order of their pattern groups generally indicate the order in which they should be used in the design process (Table 2-5). In addition, patterns chosen from the various groups have links to related patterns in the language. The highest level pattern group in their scheme is *Site Genres*, which provides a convenient starting point into the language, allowing the designer to choose the type of site to be created. Starting from a particular *Site Genre* pattern, various lower level patterns are subsequently referenced. In this way, the approach succeeds not only in providing a starting point into the language, but also demonstrates how patterns of different levels may interact with one another. However, pattern relationships are described only in general terms and their possible impact on the final design, especially in terms of usability, are not discussed in detail.

We [Javahery and Seffah 2002] have proposed a third design approach called Pattern-Oriented Design (POD). The initial motivation for POD arose from interviews carried out with software developers using our patterns from the UPADE web language. Having been motivated by the need for the development of pattern languages, the HCSE Team at Concordia University has been developing UPADE for a number of years [UPADE 2004]. Interviews carried out with software developers revealed that in order for patterns to be useful, they need to know how to combine them to create complete or partial designs. Providing a list of patterns and loosely defined relationships, as is the case for most HCI pattern languages, is insufficient to effectively drive design solutions. Understanding when a pattern is applicable during the design process, how it can be used, as well as how and why it can or cannot be combined with other related patterns in terms of usability, are key notions in the application of patterns.

POD has two features. First, it provides a framework for guiding designers through stepwise design suggestions. At each predefined design step, designers are given a set of patterns which are applicable. Second, pattern relationships are explicitly described, allowing designers to compose patterns based on an understanding of these relationships. This is in stark contrast to the current use of pattern languages, where there is no defined link to any sort of systematic process. Even if some languages such as [van Duyne 2003] provide some structure, they lack explicit

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

descriptions of pattern relationships. POD has been developed further as part of this research work, and will be revisited in subsequent chapters.

2.5. Linking User Experiences and Design Solutions

In previous sections we outlined techniques for modeling user experiences and for building conceptual designs. We will now highlight some research work done by others that attempt to provide a methodical link between user experiences and design.

2.5.1. Goal-directed design

Similar to our research work, Goal-directed design (GDD) proposed by [Cooper 1999] aims to build a tighter fit between user experiences and design solutions. GDD is an interactive design method where the primary artifact is a description of user goals. A clear distinction is made between using goals to drive design decisions instead of tasks. Goals are an end condition, while tasks are technologically-dependent intermediate steps used to achieve goals. GDD combines a number of techniques such as ethnography, market research, user models, scenarios and interaction principles. The process can be separated into five distinct phases, as illustrated in Figure 2-7.

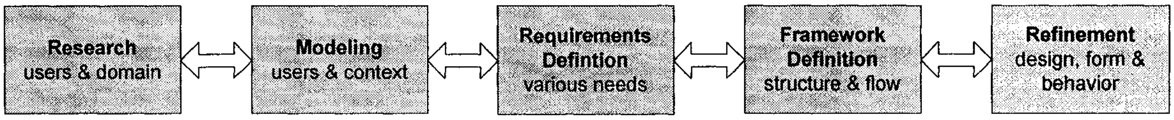


Figure 2-7: The Goal-Directed Design Process [Cooper and Reimann 2003]

A more detailed description of each phase follows:

(1) **Research**. User data is obtained using field-study techniques such as ethnography. User behaviors, along with associated goals, help with the creation of explicit user models.
(2) **Modeling**. Domain and user models are constructed using data from the previous step.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(3) **Requirements definition**. Connections are made between users, other models and the design framework. Scenario-based techniques are employed, in a significantly different manner from traditional techniques – with a concentration on goals and not tasks.

(4) **Framework definition**. An interaction framework is defined at two levels, with the hopes that it will clarify the needs as related to the system. Design principles are used and more specific design solutions are applied to classes of previously-analyzed problems. The outcome is a stable concept and logical structure of the UI.

(5) **Refinement**. Walkthrough and validation scenarios are aligned with storyboard paths through the interface and in great detail, concentrating on task coherence. The end result should be a detailed documentation of the design, form and behavior specification.

Goal-directed design, to a certain extent, addresses the difficulty of translating user and domain information into design solutions. In fact, [Cooper and Reimann 2003] claim that the Modeling, Requirements Definition, and Framework Definition phases bridge the gap between Research and Design through the use of specific techniques which bring some clarity to an otherwise vague process. For example, the use of personas is encouraged as a good user modeling technique, as well as the use of interaction design principles in the framework definition. Scenarios and resulting storyboards are also mentioned as techniques to be used in certain phases.

GDD is a good starting point to making interactive design more methodical, removing it from merely a "philosophical" view as set out in UCD. However, the information flow from one phase to another is still highly dependent on the designer's experience, and is neither traceable nor reproducible. In fact, Cooper [1999] insists that both personas and scenarios should be in narrative form, since this is the best way to communicate ideas. He observes that designers in general are not involved in the data collection phases (i.e. the Research phase), and that this results in most of the problems during design, including the "gap" between user experiences and design solutions. If designers were involved in the "field research", the gap in the design phase would be reduced. This is unrealistic, both in terms of shared expertise and system development (especially in the case of larger systems). In addition, this does not make interactive design any more systematic, but relies heavily on the designer's experience.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.5.2. Welie's Layered Model of Usability

As mentioned previously, Welie [2003] defines one of the most comprehensive UI design pattern collections. Furthermore, he proposes to describe design solutions in a format which presents how the pattern will benefit the users of the system. He argues that UI pattern descriptions should include usability problems, and that they should be categorized according to these problems. In his Layered Model of Usability [Welie et. al. 1999; Welie and Trætteberg 2000] the following four layers are described starting from the topmost: Usability, Usage Indicators, Means, and Knowledge. Figure 2-8 illustrates this model.

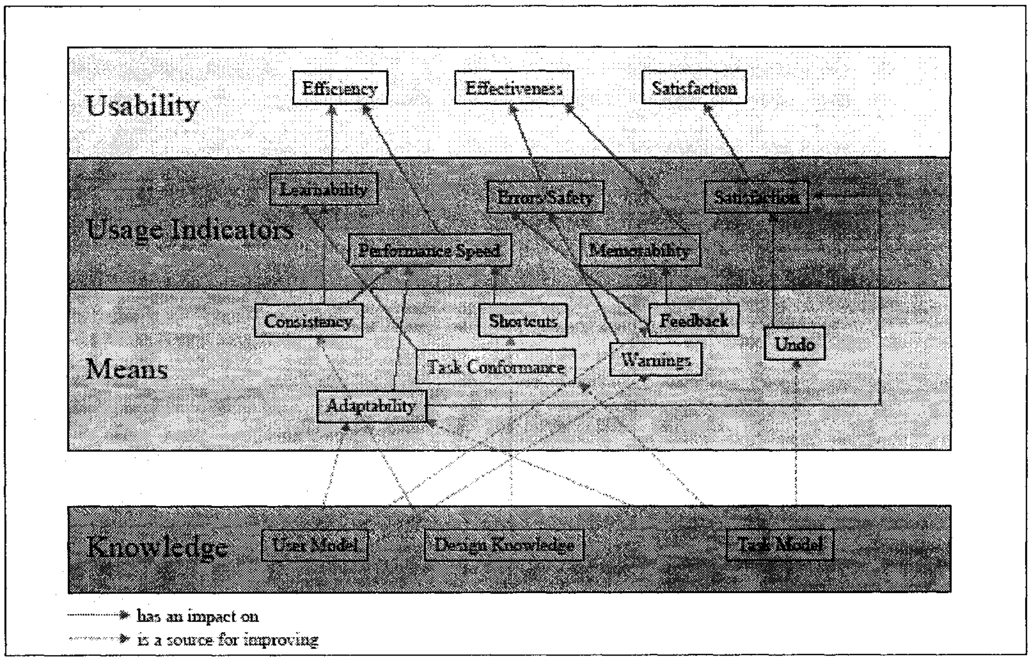


Figure 2-8: Layered Model of Usability [Welie et al. 1999]

Welie states that each pattern should affirm its impact on usage indicators, and that a pattern should improve at least one usage indicator. He discusses how patterns should somehow be linked with ergonomic principles. Furthermore, if patterns focus on usability problems of the user, they should be written following certain guidelines. Based on these guidelines, the following elements are used in describing his patterns:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- *Problem.* The problem should be related to system usage, relevant to the user or anyone else interested in usability. "Constructional" problems should not be the focus, in contrast to software engineering problems, but rather user-task oriented.
- *Usability Principle.* Solutions should be based on some usability principle. A complete set of principles is not known, but examples grouped according to Norman's [Norman and Draper 1986] user problem categories include Visibility, Affordance, Natural Mapping, and Constraints.
- *Context.* The context should also focus on the user, concentrating on context of use characteristics (user, task, and environment).
- *Solution.* The core of the solution is described, with possible pointers to other patterns relevant to the solution.
- *Rationale.* This section describes the rationale behind the pattern and the impact on usability when the pattern is applied. Often, a pattern improves some aspects at the detriment of other aspects. Furthermore, each design solution aims to provide a correct balance in the specified context. Measurable aspects of usability include performance, learnability, and memorability.
- *Examples.* The example should demonstrate the successful application in a particular system. Furthermore, a "counterexample" can be used indicating a situation where the pattern should have been applied, but was not.

By focusing on usability problems of the user, Welie, to a certain extent addresses the "gap" between user experiences and design solutions captured as patterns. However, there is little explanation given about the precise meaning of the relationships between the layers of his model, and how they may be used in common practice. Furthermore, the links to usability principles and usage indicators are not clearly defined, and their use in his pattern definitions is unclear.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.6. Summary of Related Work

This chapter provided an overview of related work. In current practice, there are various techniques to capture user experiences. These include cognitive styles, user profiles and personas. Personas are a descriptive model of the user, encompassing information such as user characteristics, goals and needs. Cooper [1999] describes a seven-step process to persona creation with intermediate steps that include identifying significant behavior patterns and designating persona types. In comparison to Cooper, Pruitt and Grudin's [2003] approach makes persona creation more rigorous with links to real data. However, there is little explanation on how designers can use personas to derive solutions for UI design. Although there is some discussion about integration of personas at different stages of software development and identifying behavioral variables, personas are still being used primarily as a communication tool. In many cases, it is hoped that the information personas contain will somehow "inspire" members of the development team to design interactive systems accordingly. Exacerbated by the fact that personas are captured in narrative form, more rigorous representations and models are needed.

In contrast to supporting techniques for capturing user experiences, existing resources for building UI conceptual designs are limited. We focus on three techniques which follow the reuse paradigm: Guidelines, claims and patterns. Guidelines and claims were previously used by designers, but patterns have been emerging as a more feasible alternative. Patterns have been introduced as a technique to capture and disseminate proven design knowledge. Their purpose is to facilitate the design of more usable systems, with a focus on the user's experience and the context of use. Erickson [2000] introduced the idea of using a collection of interrelated patterns as a lingua franca for design. Coram and Lee [1998], Borchers [2001], Tidwell [2002] and Welie [2003] have all developed pattern languages for HCI. The idea of using patterns as part of a design process has been discussed by Lafrenière and Granlund [1999] and van Duyne et al. [van Duyne 2003]. Furthermore, we proposed [Javahery and Seffah 2002] a third approach called Pattern-Oriented Design which guides designers through specific design steps and exploits pattern relationships. However, patterns are still an emerging design tool and similarly to

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

personas, they require more rigorous representations.

Linking user experiences to design solutions seems like a natural progression in HCI, but little work has been done in this area. Cooper and Reimann [2003] discuss Goal-Directed Design with personas as a central technique. They go on to identify the existing gap between user experiences and design solutions, but do little to remedy the problem. Furthermore, Welie [Welie et. al. 1999; Welie and Trætteberg 2000] makes an interesting proposition to define usability aspects in pattern descriptions. However, precise specifications on how this should be carried out are not given. As will be discussed in this thesis, we take a similar approach by focusing on usability principles as providing a methodical link between users and design solutions.

To this end, to advance the state-of-the art, it is necessary to develop more rigorous representations and models of user experiences which can be linked to conceptual designs – within a systematic and traceable process. In the subsequent chapters, we present our solution to remedy this problem, and propose a UI design method which focuses on personas and patterns as the primary design directives.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 3

Designing with Personas and Patterns: Case Study 1

This chapter presents the results of our first empirical study, with 39 end-users. The purpose of the study was for process discovery and to evaluate the feasibility of using personas and patterns as the primary design directives in narrowing the "gap" between user experiences and conceptual design. In addition, we wanted to investigate whether the development of a rigorously-defined process was substantiated. To motivate this, we developed and used a generic framework employing personas and patterns. We applied this framework to the redesign of a Bioinformatics website and performed usability tests with the new design.

3.1. Framework

We built an experimental framework to gain experience (see Figure 3-1) and as a first step to solve the research problem and test our hypothesis (see chapter 1). The essence of the framework is that personas drive the creation of a pattern-oriented design (POD). The starting point is the use of personas to model user experiences, where user information and usability evaluation results act as input. Information contained within personas is then used to identify appropriate patterns (from a pattern library) which meet user needs and fit the context of use. These patterns are used as "building blocks" and composed into a POD, resulting in a conceptual design. The conceptual design can then be used as a blueprint to build a prototype.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

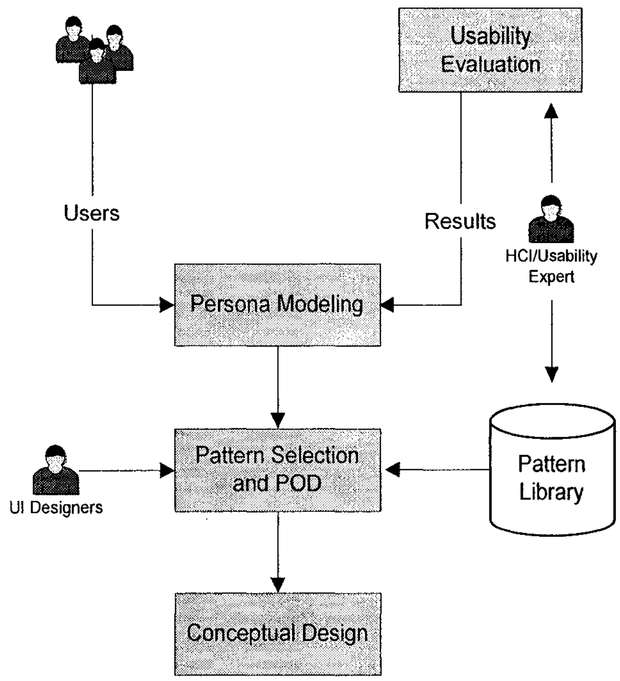


Figure 3-1: Persona to Pattern Framework

We conducted a case study with a bioinformatics information portal, where we applied the framework. The case study acted as a proof-of-concept to (1) Evaluate whether the framework results in more usable systems, (2) Evaluate validity of correlating personas with patterns while designing, and (3) Understand the limitations of the framework. It would lead us to conclude that we can either develop a more concrete method and associated process leading from personas to patterns, or that our framework needs to be refined. A summary of the experimental technique and planning steps can be found in Appendix H.

For the study, our participants consisted of 39 end-users. 16 of these users were involved in pre-design usability evaluation, to create personas and initially apply the framework. The rest of the users, 23 in all, performed comparative usability tests with the original site (design not built using the framework) and our resulting prototype (design built using the framework).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.2. Applying the Persona to Pattern Framework

3.2.1. Understanding the Domain and Users

The domain of interest for our case study was Bioinformatics. We applied our framework to the NCBI (National Center for Biotechnology Information) website [NCBI 2005], a well-established bioinformatics information portal which has access to specialized tools. The complexity surrounding the NCBI site as an interactive system is two-fold: First of all, we are dealing with a complex tool set along with a data repository of rich and critical information in a specific field of research, and with a specific user community. Secondly, the medium for dissemination of this information is the web, which has its own specificities with regards to user interaction. Users accessing and interacting with this site pursue goals ranging from simple information gathering and article searching, to using specific molecular analysis tools.

We carried out informal ethnographic interviews with bioinformatics researchers from two academic labs in Canada and a state-run research facility in France, resulting in some insightful discoveries. The NCBI site is by far the most popular bioinformatics information provider currently in the field. It provides access to nucleotide, protein and literature databases, and contains information processing methods and tools which are used even on a daily basis by researchers. However, users pointed out a number of problems with the site, including: (a) Difficulty to find desired information, (b) poor site organization, (c) information overload, (d) easy to get lost on the site, and (e) frustration arising from lengthy waiting times for receiving input. Users varied in age, goals, level of domain and application experience, profession and working environment.

The results of the ethnographic interviews, although informal, gave us enough information to progress to the next step, and come up with our first set of personas.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.2.2. Initial Persona Set

As a starting point, we used domain analysis and the ethnographic interview results to postulate representative users of the NCBI site, including information about their experiences. A biomedical expert advised us on domain-specific information. Our initial set of three personas is illustrated in Table 3-1. The main differentiating user attributes that were taken into consideration in building this set were age, work environment, and application experience. These are indicated in brackets under each persona's name. Our initial field observations indicated that these attributes would strongly influence user behavior.

First, we observed a relatively wide age range in our end-users (from young to older adults). Older users were less comfortable with site navigation. They indicated issues with cognitive and memory load, and difficulty remembering various sequences (of actions) which they had performed earlier. Furthermore, compared with other groups, older users seemed uncertain about their actions; they were more cautious about using newer technology, and required more guidance.

Secondly, we had users from industry, academic, and clinical (medical practitioner) settings. Based on our field observations, we expected variations in behavior among users depending on their work environment. For example, users from industry were driven by deadlines and time limits. They demonstrated less patience, and were looking for task efficiency and more control over the system.

Thirdly, application experience seemed to influence both the needs and satisfaction levels with the website. Basic users, who had just started to use the site within the past year, were dissatisfied. They demonstrated a sense of confusion about site structure and navigation, expressed information overload, and indicated that they needed more support. Intermediate users were more satisfied, but indicated that some of the tools they needed could "work better". Expert users were also quite satisfied, but indicated that the website was slow at times and they wanted to perform their tasks in a more efficient manner.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Persona 1	Persona 2	Persona 3
Donna Smith (Young adult, Student, Basic)	Xin Li (Mature adult, Industry, Advanced)	Dr. Thomas Johnson (Older adult, Professor, Intermediate)
She is a 24 year-old Masters student in Biochemistry. She lives with a roommate away from home. She is quite active and tries to jog daily and play soccer twice a week. She uses the internet daily for email access, and searches for biological information related to her research. She just recently started to access the NCBI site from both home and her university lab. She is using the site predominantly for information gathering, literature searches, and is trying out the BLAST (alignment) tool. Sometimes she can't find what she is looking for, and she wishes the site was less cluttered and more organized. She doesn't like asking people how to do things, but likes to figure it out on her own. She loves giving the image of being intelligent and enjoys intellectual conversation. She is a fast learner and hard worker. She often stays in her lab late on weekdays.	He is a 37 year-old researcher in a pharmaceutical company. He has a Masters in Molecular Biology. He is married with two young children. He is not very active, but tries to play tennis and squash once in a while. He uses the internet daily for e-mail, access to the company's intranet and information portal, as well as for information searches related to his work. He accesses the NCBI site weekly from his office. He is a frequent NCBI user, especially with the advanced molecular visualization tools such as Cn3D. He gets frustrated often because of lengthy processing delays when using some of the analytical tools. He doesn't really bring his work or research endeavors home, and only uses the internet at home for surfing and email. He wants to finish work as soon as possible and go home, and doesn't like to stay late.	He is a 58-year old university professor in the Faculty of Agricultural and Environmental Sciences. He holds a PhD degree in Parasitology. He is married with 3 children; all of them have moved away from home. He plays golf once a week. He uses the internet daily for e-mail access and information searches related to his research. He is an infrequent NCBI user, and only accesses the site weekly from either home or his office. Although he has been using the NCBI site and its tools for a few years, he still gets lost, which discourages him from being more active. He has a few graduate students working in Bioinformatics, and needs to stay updated on bio-computing tools and resources. He has to manage his time between teaching and supervising graduate students. The worst thing anyone can tell him is that he is not fast enough.

Table 3-1: First Iteration of Persona Set

As highlighted in the previous chapter, if constructed effectively, a persona should be sufficiently informative and engaging so that it redirects the focus of the development process towards end users and their needs. However, constructing such an effective persona is not easy. Therefore, as a means to increase their effectiveness, a persona should be supported by user and empirical data [Pruitt and Grudin 2003]. To enhance and render our personas more informative, we decided to gather more specific user and behavioral information from usability evaluations with end-users and UI experts.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.2.3. Pre-design Usability Evaluation

A number of evaluation techniques are in use for assessing user interaction behavior and usability aspects of interactive systems, including for the Web. Usability evaluation techniques include field or laboratory observation, remote testing, performance measurement (such as error rates or task efficiency), participatory design, heuristic evaluation, and the administration of objective questionnaires (psychometric assessment). Psychometric assessment and heuristic evaluation were the two techniques used in this part of the study. Further details about them, including our list of heuristics and the design of the tailored questionnaire, can be found in Appendix C and [Javahery 2003]. Please note that the first part of the NCBI study, including the pre-design usability evaluation steps, was completed as part of my Master's thesis [Javahery 2003]. In this section, we will only present a concise description of the participants and results. Note that an overview of these steps has been published in [Javahery et al. 2004].

Psychometric Assessment with End-users

Participants for the psychometric assessment consisted of 16 NCBI end-users. Unlike heuristic evaluation, small samples (e.g., N = 3 to 5) are inadequate for psychometric methods. This is because results from small samples tend to be statistically unstable and subject to sampling error. Therefore, more representative samples are desired. As a rule of thumb, there should be one subject for each item on a questionnaire. Our questionnaire had 31 items; thus, a sample of 31 subjects would have been ideal. However, due to resource limitations and the difficulty in finding samples of this user community, we were only able to include 16 participants. This is much better than the 3-5 suggested users by Nielsen, but not as ideal as we would have liked.

End-users included three medical practitioners; the rest were from four academic and industry research groups in France and Canada. Important user characteristics are illustrated in Table 3-2. Other characteristics and observations include: (1) All participants were from biomedical-related fields such as molecular biology and cancer research, with only 2 participants actually indicating that their field of research was bioinformatics, (2) Highest level of education ranged from B.Sc. to

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PhD and MD, (3) Almost half of our sample were graduate students, whereas others were researchers, physicians and bioinformatics professionals, (4) More than half of the users indicated English as their first language, while others indicated French and Chinese, and (5) Participant leisure activities included a variety of sports, traveling, and reading.

User Characteristics	Results (N=16)		
	value[3]	number	percentage[4]
Age	Mean	31.5 (SD 10.35)	–
	Min	23	
	Max	58	
Gender	Male	10	63
	Female	6	37
Application Experience	Limited	4	25
	Basic	3	19
	Intermediate	2	12
	Advanced	7	44
Internet Experience	Intermediate	1	6
	Advanced	15	94
Domain Experience	Limited	5	32
	Basic	2	12
	Intermediate	2	12
	Advanced	7	44
Work Environment	Industry	4	25
	Practitioner	3	19
	Academic	9	56

Table 3-2: Selected User Characteristics for Pre-Design Evaluation of NCBI Site

Website interaction predominantly deals with the user experience of moving through the site and interacting with all parts of the interface. Web design must incorporate what people do on the site rather than simply how it looks [Nielsen 2001]. For a bioinformatics website, we know very little about user behavior and user experiences. More consideration of all aspects of the bioinformatician's experience and interaction with the website are felt necessary, such as how the site is perceived, learned and mastered. This includes ease-of-use and, most importantly, the needs that the site should fulfill with respect to services and information.

[3] Experience defined in terms of Limited (< 6 mos), Basic (< 1 year), Intermediate (1-3 years), Advanced (>3 years).
[4] Since exact results are not necessary in our case, decimal points have been rounded up or down for easier readability.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Keeping this in mind, we analyzed information about user interaction behavior and needs. The main tasks that users performed varied extensively, and were highly dependent on their application experience. Our sample indicated that 44% of users were using the NCBI site for less than 1 year, while 56% were using the site for more than 1 year. Of these 56%, 78% were using the site on a regular basis for more than 3 years. Furthermore, the sample included both frequent and infrequent[5] users of the site. Infrequent users had similar behavioral patterns as basic application users. Table 3-3 illustrates the two emerging user groups. For the sake of simplicity, in tune with normal practice, they are called *Novice* and *Expert* users.

Group Name	User Group	Needs	Interaction behavior
Novice	Limited and Basic (<1 year experience)	Guidance Simple Navigation	Literature searching, information gathering, and basic tools (such as BLAST)
Expert	Intermediate and Advanced (>1 year experience)	Control Task Efficiency	Follow a scientific process whereby they were repeat users of specific tools

Table 3-3: Task Use and Interaction Behavior of NCBI Site Users

When we analyzed the results of the administered questionnaire, our two user groups had differing results with relation to a number of heuristics (or properties) of the site (see Figure 3-2). Heuristics that were found to be most problematic were Visibility and Navigation, Consistency and Standards, and Help.

[5] Infrequent users accessed the site rarely, once a month at most.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

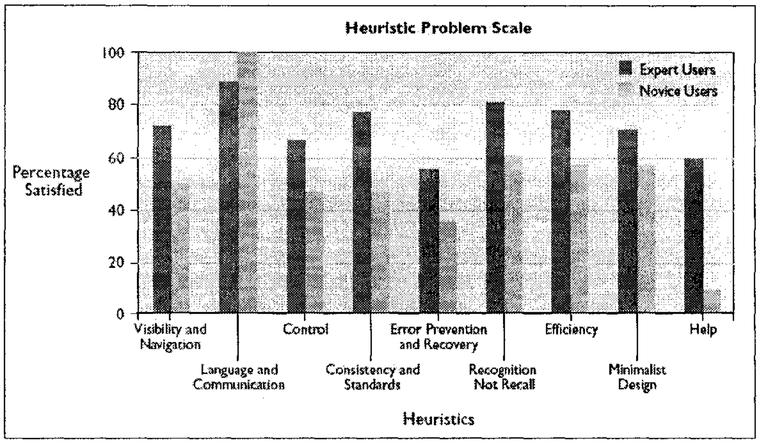


Figure 3-2: Satisfaction ratings of novice and expert users of the NCBI Web site

Heuristic Evaluation with UI Experts

A concise version of the heuristic evaluation results with UI experts are as follows: All heuristics, except for *Language and Communication*, were found to be problematic. Major problems were (1) It was easy to get lost because the path or current position in the path is unclear, (2) Difficult to get out of undesired or error states, (3) Inconsistency amongst sites, such as with different menu structures, (4) Information overload, (5) Not enough help and guidance for novice users, (6) Lack of efficient options for expert users, such as shortcuts. These findings were in confirmation with our earlier ethnographic interviews with users.

In addition, we asked 3 UI experts to comment on specific items of the site. For example, the *homepage* was found to be overloaded with links, low in visibility, and no guidance for first time users. 2 out of 3 UI experts suggested that it might be interesting to consider a different home page for different users, based on the users' experience with the site (i.e. Novice vs. Expert users). The *navigation structure* was found to be big and fairly complex, so it is easy for users to lose their orientation on the site. *Search tools* on the NCBI site were found to be relevant mainly for experienced users, but more explanation and control should be given to newer users.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.2.4. Refined Persona Set

Results from the heuristic and psychometric tests detailed in the previous section demonstrated that our initial persona set is well-founded. More precisely, all users had certain distinctive behaviors and needs that resulted in them belonging to either one group or another. However, deeper analysis indicated that one of the created personas had an overlap with the other two. Persona 3 (Table 3-1) had a majority of behaviors and needs partially attributable to one of the other two personas. Sometimes, Persona 3 was similar to Persona 1, and sometimes to Persona 2. As indicated in [Courage and Baxter 2005], we should strive to create a minimal persona set and avoid redundant overlaps, without losing essential information. Therefore, we eliminated Persona 3 since the essential behaviors and needs were already encapsulated in the other two personas.

Selected information from the refined persona set is illustrated in Table 3-4. Our most differentiating factor in terms of both interaction behavior and needs was application experience, and our refined persona set was named accordingly (i.e. *The Novice User* and *The Expert User*). However, it is important to not be misled in thinking that application experience was the only criteria used in creating our personas.

3.2.5. Selecting Patterns based on Personas

We investigated the possibility of identifying appropriate patterns based on user attributes and behavioral characteristics contained in persona descriptions. For example, in the case of the NCBI portal, a handful of patterns can be associated to the personas and related empirical studies in four aspects of web-based applications:

1. *Information architecture.* Cognitive organization of content;
2. *Screen layout.* Organization of content and actions on working surfaces;
3. *Navigation.* Interaction mechanisms;
4. *Information visualization.* Visual representations and metaphors for grouping information in cognitively accessible segments.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Welie et al. [1999] suggest categorizing patterns according to common user problems and associating them based on usability principles such as Guidance. This was a good starting point for us, but we believed that the relationship could be taken one step further, and patterns could be more closely associated with users. Let us take the example of redesigning the NCBI home page. Different characteristics and interaction behaviors of our two personas can be linked to specific patterns from the UPADE and Welie [2003] pattern languages (see Table 3-4).

Persona 1		**Persona 2**	
Donna Smith (The Novice User) 24 years old; Masters student in Biochemistry; works daily in a lab with other graduate students Needs: Guidance, Simple Navigation		Xin Li (The Expert User) 37 years old; Molecular Biologist; researcher in a pharmaceutical company Needs: Control, Task Efficiency	
Attribute/behavior	**Selected patterns**	**Attribute/behavior**	**Selected patterns**
She recently started doing bioinformatics-based research, and has only been accessing the NCBI site for 6 months	Novice user patterns	He has been accessing the NCBI site for 2 years now, and is very familiar with tools related to his research	Expert user patterns
She is still unfamiliar with all the menu options and functions and often needs guidance	On-Fly Description	English is his second language, and he is not always comfortable with spelling	Index Browsing Alphabetical Sitemap
She is still learning about the NCBI site, and actively reads "General NCBI Information" and "About NCBI"	Executive Summary	He uses the NCBI site for specific tasks, such as secondary structure prediction for proteins and wants to save his results	Shortcut MySpace (customized)
She uses the site mainly for literature and article searches (such as Pubmed), educational and information-gathering, and has only started to do sequence alignment searches	Index Browsing Simple Search	Likes to limit his searches to specific species and doesn't have patience to go through a long list of possibilities	Advanced Search
She gets lost looking for information after advancing more than 3 layers, and needs to go back to a safe place	Convenient Toolbar Dynamic Path	Likes to know about recent discoveries and advances in the field	Teaser Menu Executive Summary

Table 3-4: Selected Information from Personas and Associated Patterns for Study 1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.2.6. Composing Patterns

After pattern selection, we used the POD approach (see chapter 2) to combine appropriate patterns based on the desired user-task and user behavior. Pattern-oriented design may be used to create web applications that customize the solution of common problems for different personas. Alternatively, they can guide designers in finding compromise solutions based on design trade-offs. One ideal design strategy in this case would be the implementation of separate home pages for novice and expert users. The two different home pages would actually be the result of using different types of patterns for each group of users. Although this strategy may be ideal for the user community, our context information revealed that its practicality and maintenance would be limited for such a large and complex website, with a site architecture that is so multifaceted. It therefore made more sense to settle for a compromise, and try to make the site usable for both types of users.

Taking into account earlier evaluation results, the two personas and the original NCBI site, one example of a design decision is as follows:

> "The left navigation menu on the homepage has important site links, however a textual description of the link is provided under each name; this is only useful for novice users who don't know what kind of information they can find from the links. This convenience for the novice user is a hindrance for an expert user as it contributes to more scrolling."

A solution would be implementation as rollovers (on-fly description pattern) to help new users. Using the selected patterns, we built a new design for a subset[6] of the NCBI site – this included the portal homepage, main navigation elements, site map, search tool, and some content pages. Selected pattern compositions for navigation elements are illustrated in Figure 3-3 (as a pattern skeleton) and in Figure 3-4 (the prototype). Further details, including additional examples of design decisions and a comparison of the old and new designs, are illustrated in Appendix C.

---

[6] The NCBI site is considered a large website, with over 5000 pages. A subset was chosen for redesign.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

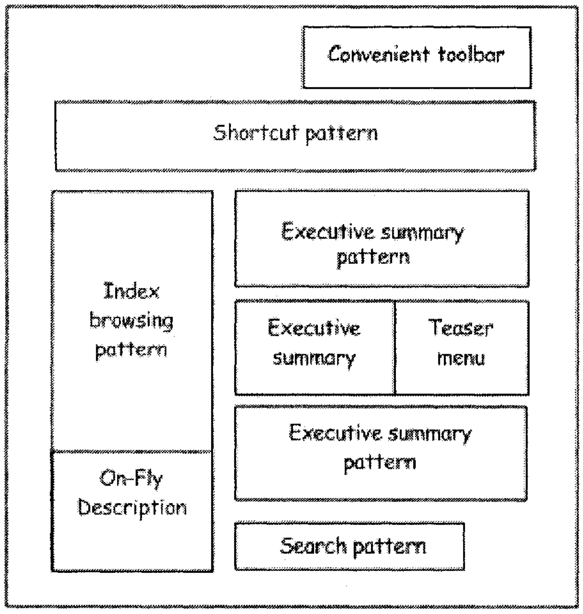


Figure 3-3: Pattern Skeleton of the NCBI Homepage

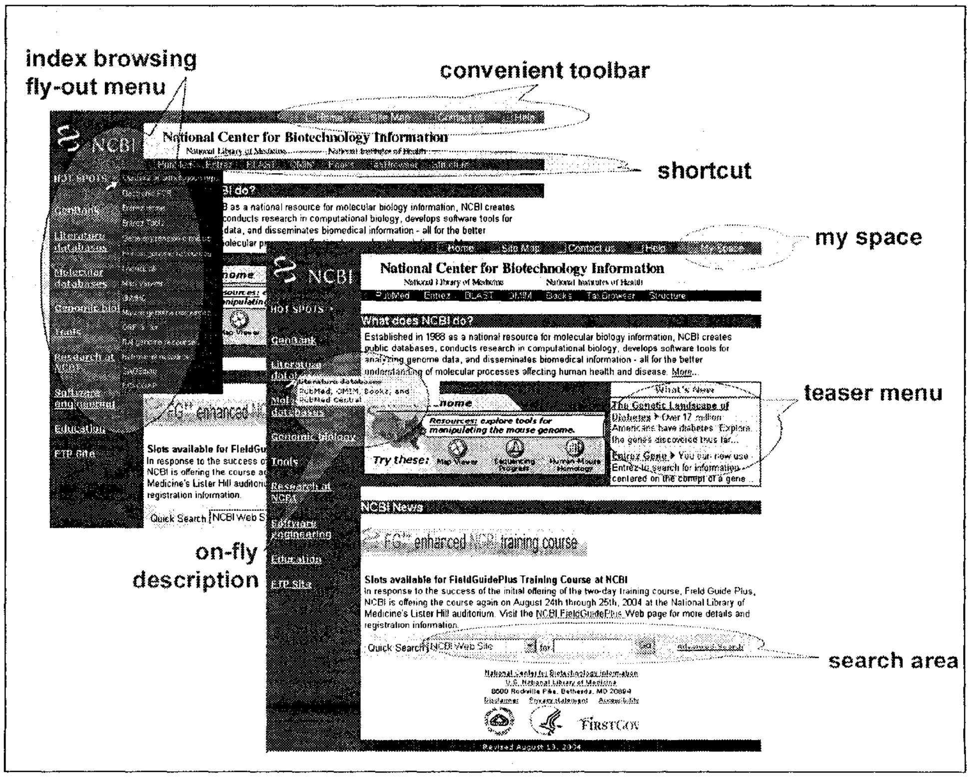


Figure 3-4: Pattern-Oriented Design of NCBI homepage

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.3. Post-Design Usability Evaluation

After redesigning the homepage of the NCBI site by following the proposed framework and choosing appropriate patterns, we evaluated the new design (using a functional prototype) in terms of usability. We used principles of software usability measurement based on ISO standards, and as indicated in [Abran et al. 2003]. We conducted a comparative study with the current NCBI site. According to [NCBI 2005], the site has been designed following "usability and user-friendly design guidelines".

3.3.1. Participants

Our participants this time included 23 users; 19 users who fit our *novice* persona (see Table 3-5), and 4 users who fit our *expert* persona (see Table 3-6). The set of participants were selected initially based on a phone interview, and were not the same users as during pre-design usability evaluation. Users were differentiated based on application experience, where novice users had limited or basic experience, and expert users had intermediate or advanced experience.

Design	Age		Sex		Computer Experience*		Domain Experience*	
	mean	SD	M	F	Mean	SD	Mean	SD
original	26.67	5.43	4	5	3.44	1.33	2.44	1.13
new	27.70	6.18	6	4	3.70	1.16	2.50	0.85
totals	27.21	5.70	10	9	3.58	1.22	2.47	0.96

Table 3-5: Aggregate description of novice participants[7]

Age		Sex		Computer Experience*		Domain Experience*	
mean	SD	M	F	Mean	SD	Mean	SD
30.50	4.12	4	0	4.25	0.96	4.25	0.96

Table 3-6: Aggregate description of expert participants

Novice users came from a variety of educational and professional backgrounds (see Table 3-7). Since novice users were tested with both quantitative and qualitative measures, a larger sample

[7] *Rated on scale from 1 to 5, where 1 is basic; 5 is expert.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

was required. Expert users were all from a bioinformatics background, consisting of graduate students (2), a researcher (1), and a professor (1).

Engineers	6
Bioinformatician / Biologists	5
Computer Scientists	3
Students (arts and science)	2
Others[8]	3
total	19

Table 3-7: Background of novice users

3.3.2. Method

An overview of the test protocol is illustrated in Figure 3-5. For novice users, we used a *between-subjects* (that is randomized) design, where each participant was assigned to a different condition [Dix et al. 2003]: (1) experimental condition which was the pattern-oriented design, and a (2) control condition, which was the original design. On one hand, by using a between-subjects protocol, we were able to control any learning effects which would have occurred if each user were tested both designs. On the other hand, this type of protocol required a greater number of participants and a careful matching of participants between the two conditions; the reason being that individual differences between users can bias the results.

[8] Comprises business (2), and environmental design (1) professionals.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

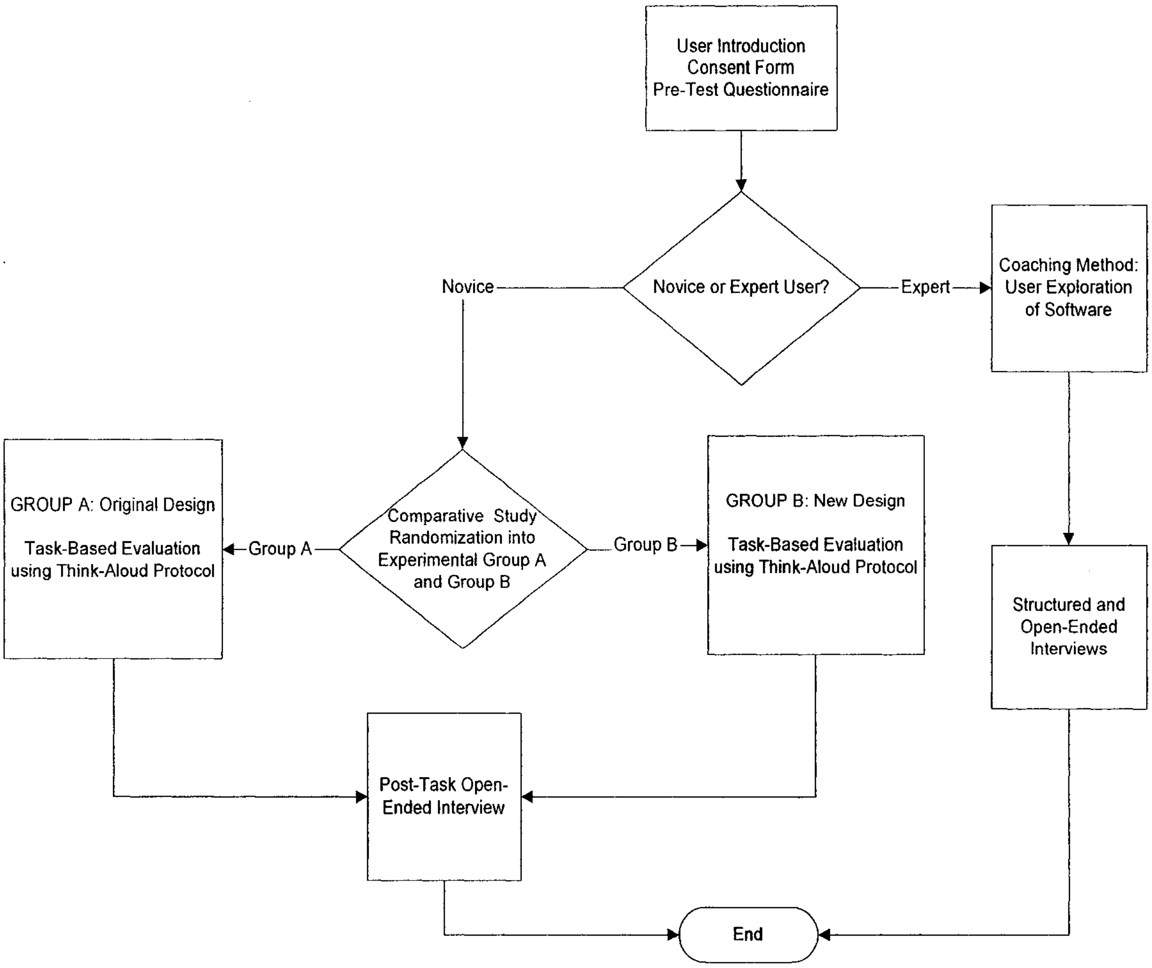


Figure 3-5: Testing Protocol for NCBI Study Post-Design Usability Evaluation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Quantitative testing, as task-based evaluation, was conducted with novice users. These users were given four common and basic tasks to perform on the website, with the purpose of calculating task duration and success/failure rates. Their performance was logged, timed, tested and analyzed by using the Morae Usability Testing Tool [Morae 2006] (see Figure 3-6). The sessions were also videotaped and any facial expressions were recorded. The think-aloud protocol was used to allow users to express themselves orally during the test, and for us to gather further information.

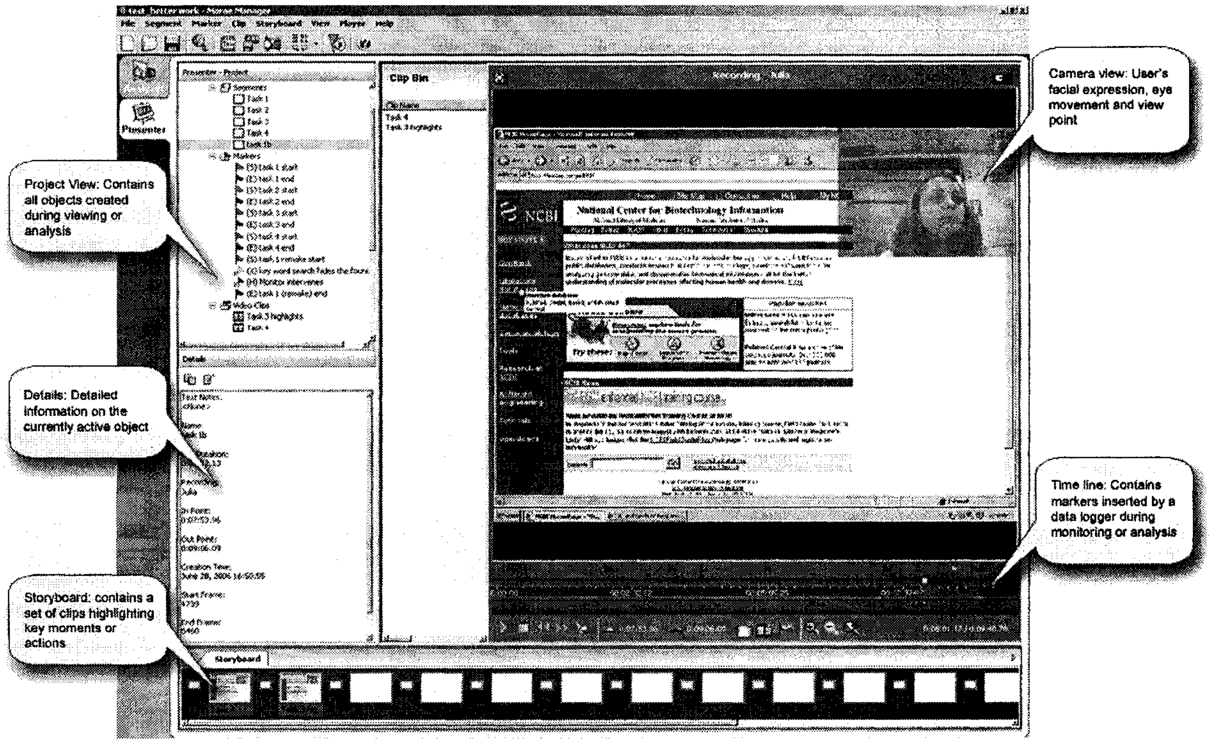


Figure 3-6: Morae Usability Testing Tool

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Four common tasks were chosen and designed with the advice of a biomedical expert. A complete list of tasks can be found in Appendix D. An example task is presented below:

> **Scenario:**
> You are a graduate student in Human Genetics. You have obtained some new protein sequences, and you need to view their 3-dimensional structure using the Cn3D utility.
>
> **Method:**
> For purposes of this exercise, you will be asked to verbalize each step performed as you are performing the task.
>
> **Task Steps:**
> 1. Start at the NCBI Homepage.
> 2. Find the link that allows you to download the Cn3D utility.
> 3. Stay on the current page.

In addition, satisfaction ratings were taken. Results were compiled based on the post-task interview (which was open-ended), as well as feedback obtained both during (users were *thinking aloud*) and after the task-based evaluation. In the latter, users were asked to comment about their satisfaction with the design after performing the four required tasks. The post-task interview included questions about likes and dislikes of individual web pages including the portal homepage, the overall site, ease of navigation, and possible improvements.

Since expert users already had extensive experience with the NCBI site, we performed only qualitative evaluations with them using both structured and open-ended interviews. They were given time to explore both the original and new versions of the website, and were asked a set of questions based on their experience. They were asked to give first impressions about both sites (likes/dislikes and any noticeable differences), as well as answer specific questions about individual web pages (such as the portal homepage), the overall site, and ease of navigation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.3.3. Results

Task duration, failure rates, and satisfaction measures were collected and analyzed as usability indicators for the comparative study with novice users. Task duration and failure rates are quantitative, while satisfaction is qualitative (and in this case, used as a discrete measure). We also performed post-test interviews with expert users to determine their satisfactions and concerns with the original and new site.

We used ANOVA tests to assess if the mean values obtained are significantly statistically different. The Analysis of Variance (ANOVA) is a statistical test that compares the amount of variance between groups to the variance found within groups. In addition to statistical significance, we computed effect size, eta-squared ($\eta 2$), which is a measure (of the magnitude) of the effect of a difference, independent of sample size. In HCI, due to commonly small sizes of studies, effect size has been found to be appropriate [Landauer 1997]. In general the greater the value of $\eta 2$, the greater the effect. Some HCI practitioners [McGrenere et. al. 2002; McGrenere 2002] use the following metrics for interpreting eta-squared: .01 is a small effect; .06 is medium; and .14 is large. In our analysis, we compare the ANOVA results and size effect in order to come to a conclusion.

Quantitative Results

Task duration results are illustrated in Table 3-8. We used ANOVA single factor tests to compare task times of Designs O (original) and N (new). Our hypothesis for this test was that we would have a significant improvement in the time required to complete a task in Design N versus Design O. Five tests were performed. Tests 1 to 4 were performed by comparing the time required to complete a task for each design. Each test compared values within a given task. Test 5 was performed by comparing the total time required to complete all four tasks for each design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Task #	Mean times (original)	Mean times (new)	F	p	η^2	Average time difference (in %) [9]
1*	4.24 (SD 2.42)	1.92 (SD 1.80)	4.75	0.05	0.25	-54.80
2	2.79 (SD 1.98)	0.85 (SD 0.29)	8.47	0.01	0.35	-69.72
3	4.92 (SD 3.24)	1.61 (SD 0.85)	8.77	0.01	0.35	-67.26
4	3.12 (SD 1.11)	1.82 (SD 0.91)	7.43	0.01	0.32	-41.67
all	14.44 (SD 3.88)	6.41 (SD 1.86)	27.92	0.00	0.67	-55.61

*All F values have degrees of freedom (1, 16) except task 1 and "all" where it is (1, 14).

Table 3-8: F values of ANOVA single factor tests when comparing task times

The results ($p \leq 0.05$) indicate a significant improvement in both individual and total task times with the new design. These results suggest that there is a statistically significant relation between the time required to complete a task and the type of design used. Moreover, there was an improvement of at least 40% when considering the average time required for completing a task. At the same time, if we consider total task time, we note that overall improvement was more than 55%.

Failure rates were also analyzed. Our hypothesis for this test was that there should be a significant improvement in failure rates with Design N versus O. However, the ANOVA single factor tests did not show any relation between the type of design and failure rates. As we can see in Table 3-9, the tests for task 1 and 3 indicated that there is no statistically significant effect of Design N on the failure rates.

Task #	F	p	η^2
1	2.29	0.15	0.13
2*	N/A	N/A	N/A
3	2.29	0.15	0.13
4*	N/A	N/A	N/A
all	6.4	0.02	0.29

* No failures were detected for task 2 and 4 in either design

Table 3-9: F values of ANOVA single factor tests when comparing failure rates

[9] The average time difference (in %) was calculated using $\frac{mean_N - mean_O}{mean_O} * 100$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Let us take a closer look at the available data. In Design O, 4 out of 9 users failed one task; none failed more than one. In Design N, no failures occurred. On average, there were 0.44 fails per user with Design O, and none with Design N. If we take a look at the overall failure rates of users, we find that F from the ANOVA test and η^2 values show that there is a relatively strong, positive and statistically significant effect of Design N on the failure rates: F (1, 16) = 6.4, $p<0.05$, $\eta^2 = 0.29$. From the five tests (tasks 1-4, and *all* tasks), only one confirmed our hypothesis. Therefore, we cannot affirm that Design N offered a significant improvement in failure rates. Further investigation by increasing the number of participants may provide additional insight and confirm or reject our hypothesis for this particular test.

Qualitative Results

Satisfaction measures with novice users were subjective measures based on a five point scale (1 = totally dissatisfied and 5 = completely satisfied) with the design. Our hypothesis for this test was that Design N would have a significant improvement in satisfaction ratings over Design O. Results of the test (F (1, 16) = 11.53, $p < 0.05$, $\eta^2 = 0.42$) suggest that there is a significant difference in the satisfaction ratings between the N and O designs. Moreover, when we consider average ratings of both designs (N = 3.44, SD = 1.13; O = 1.89, SD = 0.78) we find that overall, the users were almost two times more satisfied with Design N as compared to Design O.

It is also interesting to note that the standard deviation in the case of Design O is 0.78, which indicates that most of the users had a similar opinion: In general, they were dissatisfied. On the other hand, the new design has spawned much more variation (SD = 1.13) in opinions: 45% users were satisfied with the design (ratings of 4 and 5), 22% were dissatisfied (rating of 2) and 33% were divided (rating of 3). These results may suggest that some users noticed certain issues in the original design that were not modified in the new design, or that were introduced during the redesign. It is important to also recall that only sub-parts of the current design were redesigned. Further analysis would be required to see if there is some correlation between satisfaction ratings for these users and some characteristics in their personal data. To summarize, although our results

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

indicate a higher satisfaction rate with Design N, standard deviation of the opinions indicates that more analysis should be performed in order to search for the remaining issues which trigger dissatisfaction.

Post-interviews with expert users consisted of structured and open-ended interviews. Results varied in response. Two out of four users preferred Design N, and noted that it was more "simplistic" and "lightweight". Other comments included that the new design introduced less information overload and increased clarity. They also found it easier in terms of navigation, although they commented that they would need to get used to the new site. The other two users did not have an overall preference, but preferred different aspects of the two designs. The first user found Design N visually more appealing and "easier on the eyes". However, he was having a difficult time performing his usual tasks due to the reorganization and changes in navigation elements. Therefore, for performing his tasks, he preferred Design O because he was more "comfortable" with it. The second user did not find much of a difference between the navigation structures of the two designs. In terms of visual representation, he preferred Design O because it was more "compact" and Design N included too much white space, which he did not appreciate. In terms of functionality, he preferred Design N because he was able to perform some of his typical tasks more rapidly, and had an easier time searching for information.

Overall, the qualitative results with experts were what we expected. Expert users have been using the site for an extended amount of time, and have become used to certain elements and visual representations. In addition, they have become habituated in performing a task using a specific set of steps, and comfortable with following a certain navigation path. However, saying that, the results although mixed, were still somewhat positive. Further testing and analysis would be needed in determining how to best accommodate this group of users with the new prototype.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4. Summary of Case Study 1

In this chapter we reviewed a case study that provided us with an experimental infrastructure to test our ideas about relating personas and patterns. We proposed and used a design framework based on UCD principles, which aims to narrow the existing "gap" between user experiences and conceptual design. The framework is based on personas and patterns as the primary design directives, with a focus on empirical studies and experiments with real users during the pre-design stage. We believed that linking these two techniques within some kind of framework would be useful for designers. And, even more so for novice designers, who often have limited knowledge of usability principles and user-centered design techniques. Our premise was that a practical process for designers with clearly-defined phases would yield User Experience (UX)-based designs, and increase product usability – all of this without necessitating designers to have years of expertise in areas such as user modeling or usability engineering.

We applied this framework to a design project with a Bioinformatics website, as a proof-of-concept and for process discovery. In particular, the goals of this study were to evaluate: The effectiveness of correlating personas and patterns while designing, whether the framework results in more usable systems, and its limitations.

First, we found that applying the framework facilitated our design activities, allowed us to incorporate sound UCD principles into our design, and afforded guidance to an often ad-hoc process. In other words, it provided us with some structure. Since the starting point was creating personas, the focus of the design activity was directed to the users early on. Furthermore, personas are a relatively lightweight user model, and we did not require a user or cognitive modeling specialist for their creation. By developing personas iteratively using empirical evidence, it allowed us to determine more precise interaction behavior and determine usability problems with the application; these points were essential in selecting HCI patterns. In this vein, the framework follows the reuse paradigm through the use of these patterns, enabling us to make design decisions based on best practices. Notably, in current practice, there exists no commonly agreed upon UI design process that employs patterns and their languages as first class tools. It was our intention to further develop the framework to overcome this problem.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Secondly, after applying the "Persona to Pattern" framework to the NCBI site, we carried out comparative usability studies with the original site and our prototype. We wanted to evaluate if the framework resulted in more usable systems. We used principles of software usability measurement based on ISO standards, as indicated in [Abran et. al. 2003]. The results were positive for both quantitative and qualitative measures. In particular, our prototype indicated a statistically significant decrease in task duration and an increase in satisfaction with novice users. For total task time, we noted an overall improvement of more than 55%. Moreover, when we considered average satisfaction ratings of both designs, we found that users where almost two times more satisfied with our prototype as compared to the original design. As expected, our qualitative results with expert users were also positive but more mixed, since they had extensive experience with the original site.

Thirdly, there were some limitations we needed to address. The framework was a first step in using the techniques of personas and patterns together. We noted that links made between user experiences and design solutions were based on narrative and qualitative data, assessed manually where the "best" pattern within a specific context was selected. Any further development of our framework should include identifiable and discrete steps, and not be subject to extensive interpretation by the designer. This would require some formalization of the information contained in both personas and patterns; which we will discuss in subsequent chapters. We also realized early on that we would refer back to the personas for additional information both during the selection of appropriate patterns, and for pattern-oriented design. At times, the amount of additional information contained within personas was lacking. Therefore, an enhancement of persona descriptions with interaction behaviors, scenarios, and goals would be an added-value in guiding designers during design decisions.

Our preliminary results were encouraging. This led us to conclude that a more concrete process leading from personas to patterns was substantiated.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 4

Proposed Design Method

In this chapter we propose a novel UI design method. We first review the main phases of the method and relate them back to the experimental design framework used during our first case study. We then discuss key principles that have been taken into consideration for its development.

4.1. Overview

The NCBI study, discussed in the previous chapter, provided us with an experimental framework to test our ideas about relating personas and patterns. In this chapter we reuse the acquired experiences and expertise in order to propose and document a novel UI design method. As portrayed in Figure 4-1, the method distinguishes between the following three phases:

1. Persona Creation, where designers create personas based on real data and empirical studies.
2. Pattern Selection, where certain attributes and behaviors from the first phase drive the selection of candidate patterns for the desired domain and context of use.
3. Pattern Composition, where designers use a subset of the candidate patterns as building blocks to compose a conceptual design.

It is to be noted that designers are free to repeatedly refine the artifacts produced at each phase before proceeding on to the next phase thereby incorporating changes in personas, selected patterns, or pattern compositions.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

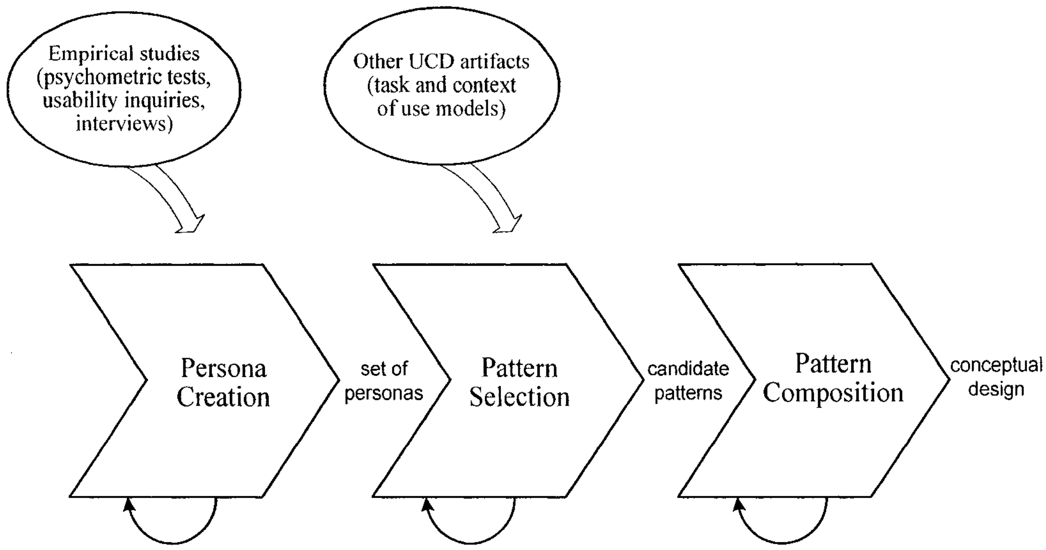


Figure 4-1: Proposed Design Method

Two additional sources of information contribute to the above phases, (1) Empirical studies and (2) Other UCD artifacts. Examples of their use were illustrated in part by the NCBI study. First, empirical studies such as ethnographic interviews provided the groundwork for a better understanding of users and their needs, resulting in the first set of personas. Furthermore, usability inquiries were useful for eliciting information about interaction behaviors and usability-related design issues. In particular, we used heuristics and psychometric evaluations to gather additional information for persona creation. This information also fed into our design decisions. Alternatively, we could have carried out inquiries with a similar application. Secondly, other user centered design artifacts besides personas and patterns are necessary in creating an overall design. User-task, context of use, and interaction models provide essential information during any design process and are important guides for establishing UI structure and system-related behavioral details. As an example for the latter, we used decision trees to describe the choices faced by a user during interaction with the NCBI site.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.2. Key phases

Each phase results in the production of one key artifact, which is then used in the next phase. Persona creation yields a set of personas; pattern selection, a set of candidate patterns; and pattern composition, a conceptual design.

4.2.1. Persona creation

Any design initiative should first focus on user experiences. By understanding and analyzing users and their behaviors, we can build personas for the target user community. In our method, in addition to acting as a communicative tool, we use our personas as an artifact which has direct implications on the conceptual design. We describe the following activities for persona creation:

1. **Understand the users.** An initial understanding of users based on domain analysis and field studies are necessary. This activity results in a detailed description of user attributes. In the NCBI study we conducted ethnography and informal interviews to better understand the users and their interaction with the web application. We noted down variations in user attributes, needs, and current usability problems with the site. Examples of attributes included computer and application experience, domain knowledge, profession, age, and learning style. Furthermore, we noted information about interaction behavior with the site, such as task-based needs and attitude towards features. This information acted as input into creating our first persona set.

2. **Group users to create initial persona set.** Users are grouped based on most important attributes. Each group is used to create a typical user, resulting in the first set of personas. For example in the NCBI study we first grouped users based on what we believed to be the most important attributes: Age, work environment and application experience. We chose these attributes because our initial analysis indicated they contributed to differences in interaction behavior. For example, older users were less comfortable with site navigation, industry users were driven by deadlines and demonstrated an increased need for control, and novice

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

application users exhibited information overload.

3. **Perform empirical studies**. The resulting persona set may be larger than needed and not the most optimal. Therefore further empirical user studies help in determining the most significant user groups and in identifying user attributes captured in the corresponding personas. For example, our persona "Xin Li" seemed to have similar behaviors to both our other personas, in our particular context of use with the NCBI site. At this point, the designer has a better idea of the users, and a set of participants can be identified for further empirical studies. The type of study depends on the availability of an older version of the application, prototype, or similar application. Studies can include psychometric tests, usability evaluation and interviews. Results from these studies will be used to both enhance the persona set, and will feed directly into design decisions.

4. **Modify and enhance personas.** The set of personas is modified and enhanced based on information from the empirical studies. In particular, acquired insights about attribute dependencies and interaction behavior are included in the persona descriptions. Enhancements may also include scenarios, which are stories about the persona in a specific context of use. They typically include information about the individual user, the task or situation, the user's desired outcome or goal, task flow details, timeline, and envisioned features. Initial steps towards this direction were undertaken during the NCBI study by including user needs and interaction behaviors in the persona descriptions.

The set of personas is iterated as many times as needed to fit the context of use. During the various iterations, personas may be added or omitted. Additions occur when designers need to include an important attribute which will be conflicting if added to the existing personas. Omissions occur when two personas vary in attributes that are not important for the designer to capture. Therefore, only one of the two personas is kept.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.2.2. Pattern selection

The next phase of the method is concerned with the selection of candidate patterns based on the discovered persona specifications. The information captured within personas is used to directly derive design solutions. This entails finding associations between user attributes within the personas and the forces which constitute a pattern.

Before candidate patterns can be selected, the designer needs to choose an appropriate pattern library. Pattern libraries are typically organized according to domain; examples being patterns for visualization [Wilkens 2003], web [Welie 2003; Javahery and Seffah 2002], mobile [Welie 2003, Javahery et al. 2003], and GUI [Tidwell 2002]. For the NCBI study we used patterns for Web applications, namely the UPADE and Welie pattern languages. The next step consists of associating patterns with users and their needs. HCI patterns include valuable information about users, their experiences, as well as usability and design principles. This information is typically included in the *context* or *forces* attribute of the pattern.

Based on the context information entailed in pattern descriptions, we can draw associations between certain user categories and patterns. Examples include patterns for color-blind users, novice users, children, and users with disabilities. An example of a pattern for a novice user, the wizard pattern, is illustrated in Table 4-1. We note that the wizard pattern was considered during the NCBI study as a candidate pattern addressing the novice user group. We can also establish a more complex association between user needs and usability principles. From persona descriptions, we derive information about needs (examples being a user's *need* for guidance) and associate them with usability principles which are to be inferred from the pattern description. The wizard pattern also addresses the *guidance* usability principle which in turn satisfies the user's need for guidance.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

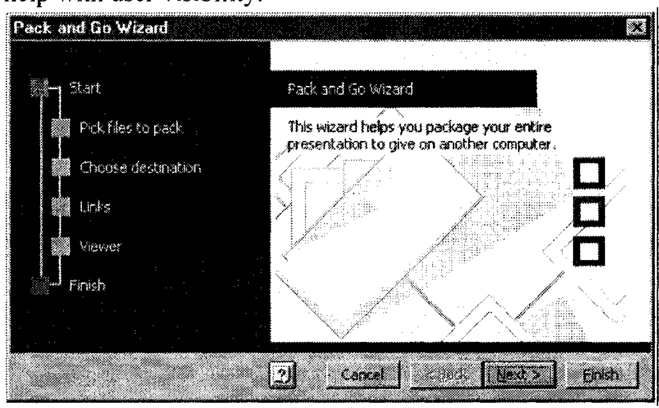
Title	**The Wizard**
Context	This pattern can be used when a **novice user** needs to perform an infrequent complex task consisting of several subtasks in a linear order where decisions need to be made in each subtask. The number of subtasks must be small, e.g., typically between 3-10.
Problem	The user wants to achieve a single goal but several decisions need to be made before the goal can be achieved completely, which may not be known to the user. A guiding principle here is that the user needs guidance.
Solution	**Take the user through the entire task one step at the time. Let the user step through the tasks and show which steps exist and which have been completed.** When the complex task is started, the user is informed about the goal that will be achieved and the fact that several decisions are needed. The user can go to the next task by using a navigation widget (for example a button). If the user cannot start the next task before completing the current one, feedback is provided indicating the user cannot proceed before completion (for example by disabling a navigation widget).
Examples	The user wants to package a presentation so that the presentation can be given on another computer. Several relevant decisions need to be taken and the wizard helps the user take these decisions. The current position in the task flow is highlighted during each step, to help with user visibility. 
Related Patterns	Two Panel Selector, Titled Sections, Responsive Enabling, Responsive Disclosure, Good Defaults

Table 4-1: Wizard Pattern for Users with Guidance Needs [Welie 2003]

The set of selected patterns should be iterated as many times as needed, until the designer is satisfied with the pattern set. As further design information is synthesized, patterns will be added or omitted from the selection.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.2.3. Pattern composition

During this phase, a pattern-oriented design (POD) is generated by composing the set of selected patterns. A valuable advantage of patterns and their associated languages are their generative nature, meaning that they can essentially be combined together as building blocks, even "plugged into" an overall structure, resulting in a comprehensive design. In order to effectively apply patterns, we need to have an understanding of when a pattern is applicable during the design process, how it can be used, as well as how and why it can or cannot be combined with other patterns. We therefore define two essential elements related to composing a pattern-oriented design: (1) The use of a POD model for design structure and as a guide in stepwise design decisions, and (2) exploiting pattern relationships for incremental design generation.

First, the POD model holds information about the overall design structure, including a breakdown of the structure into different UI facets [Vanderdonckt et al. 2003]. For example, in UPADE, a website is organized according to architectural, structural (page managers and information containers), and navigation support patterns. Such a POD model can act as a guide in stepwise design decisions, where patterns are composed according to each facet separately and then combined in an overall design. Secondly, we should exploit relationships between patterns. One of the pattern attributes is "Related Patterns", which includes alternative or complementary patterns that we may want to consider as part of our design. Table 4-1 depicts that the Wizard pattern is related to the *Responsive Disclosure* pattern; indicating that the display of a step is delayed until the user finishes the previous step [Tidwell 2005]. Pattern interactions and dependencies are very useful, contributing to an incremental generation of the design.

In the NCBI study, we used the POD model to make decisions about which patterns to choose for the new site design. For example, Figure 3-4 illustrated our choice of navigation support patterns. Furthermore, some of these patterns were chosen as a direct result of exploiting pattern relationships from the initial list of selected patterns.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.3. Key principles

Our design method is based on a set of key UCD principles which we have enriched with "engineering-like" concepts such as reuse and traceability. Our method is based on the following five key principles: (1) a focus on all user groups and their needs; (2) the incorporation of behavioral rationale; (3) the systematic and traceable application of gathered knowledge; (4) an embracing of reuse; and (5) taking a lightweight and pragmatic approach. In what follows, we explain each of these principles in greater detail:

4.3.1. Focus on *all* user groups and their needs

We consider all user groups in persona creation. As an illustration, one of the strengths of using personas is that we do not forget about boundary cases of users, which are individuals who may not constitute a large percentage in terms of market segmentation, but whom are still primary users of the application. As an example, seniors often fall into this category for e-commerce applications since they are more skeptical about using technology for their transactions. Although they may constitute a smaller percentage of the end-users, they are still an important consideration when designing the application.

We also consider special categories of users, referring to individuals who have "special" needs in terms of the user interface design, distinguishing them from the rest of the population. These include individuals who are children, color-blind, have low literacy, or even novice users. They have specific needs, and a set of patterns is chosen accordingly. At the same time, design solutions are chosen based on certain usability and user-centered design principles. This ensures that the user's needs strongly influence the pattern selection phase.

4.3.2. Incorporation of behavioral rationale

A cardinal responsibility of HCI is bringing behavioral sciences such as psychology, sociology, and anthropology to bear upon interactive system design. One problem is that designers generally

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

have limited expertise and formal training in the behavioral sciences. Therefore, this crucial aspect of design is often neglected. Our method provides a multidisciplinary approach which incorporates behavioral rationale into design without requiring the designer to have extensive expertise in the field. Furthermore, it proposes a possible bridging representation between behavioral rationale and design recommendations by linking personas to design patterns.

The resulting coupling of best design practices with the insights of HCI, augmented by knowledge drawn from psychology and the behavioral sciences, puts much power into the hands of the designer. Because its inner workings occur in the background, the developer's mind is not cluttered with bookkeeping details as it might be in other user-centric methodologies.

4.3.3. Systematic and traceable application of knowledge

We systematically apply scientifically-gathered user knowledge in design practice. Our method provides three clearly-defined phases. Depicted or portrayed in isolation, each phase is encapsulated and can be seen as a self-contained method, with its own activities. The outcome of one phase feeds into the next phase. For example, the resulting artifact from Persona Creation (phase 1) is a set of personas, which are then used during Pattern Selection (phase 2) as a basis of choosing relevant patterns for design.

Furthermore, the outcome of our method is traceable. Jacobson [1992] defines traceability as "to trace objects in one model to objects in another model". In our case, we are dealing with the user and design models; our method allows designers to determine why they ended up with a particular design, and trace back their steps to persona creation. To illustrate, our conceptual design may feature a specific pattern – designers can figure out why this pattern was chosen, based on which user and what kind of user information.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.3.4. Embracing Reuse

It is widely accepted that HCI reuse should be more prevalent through effective means and artifacts that embody sound theory. HCI has had a peculiar connection with reuse. In the past, UI developers employed guidelines and standard reusable components where UI reuse stopped at the surface. Respective examples include narrative descriptions of UI elements such as toolbars, and commercially available message boxes and buttons. Functionality or more complex dialogue was rarely addressed in either case [Sutcliffe 2000].

We follow the reuse paradigm in our method by employing patterns to drive design solutions. Patterns somewhat remedy the HCI reuse problem; they hold a significant amount of knowledge in terms of HCI and user-centered design principles, provide a firm context of application for their use, and have the potential of being extended to include further information about usability principles. In especially the past five years, patterns have been widely being accepted as ideal vehicles for the reuse of successful design solutions in HCI [Sinnig et. al. 2004].

In practical terms for designers, patterns are employed as reusable design blocks which apply in particular situations. To be an effective HCI reuse artifact, besides containing useful knowledge, designers have to be able to effectively access pertinent information for their designs. We address this retrieval problem by proposing a context for pattern reuse through the use of specific user attributes and needs from the user model, which can act as a sort of "indexing mechanism".

4.3.5. Light weight and pragmatic approach

Studies indicate that although usability engineering is deemed as an important undertaking in industry, the actual practice of HCI is not common -- lack of resources and expertise, complex methodologies including the use of heavyweight models, and the unclear delivery of HCI knowledge from theory to practice are amongst some of the concerns plaguing the field. For example, cognitive models have been used in HCI in the past. In addition to being difficult to use, a predicament with these models is how to include the necessary detail in a way that allows them to still offer useful predictions to designers [Muller and Czerwinski 1999, Sutcliffe 2000].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Our approach is lightweight and pragmatic. We do not consider all context factors – but rather, the most relevant user information as an input to our method and the most important subset of behavioral and usability factors that could influence design. We believe that heavyweight approaches and complex methodologies do not meet current requirements of designers in terms of required expertise, time and flexibility. As an illustration, many personalization systems take a heavyweight approach and extensively model users, their preferences and tasks. Heavyweight models are exceptionally taxing to construct and adapt – when sufficient resources are allocated, they can be successful. However, time constraints and other limitations can force the designers to simplify the process, thus reducing the fidelity of the model [Konstan 2001].

Our alternative is to use a less complex model of the user, and deliver useful predictions via examples of good practice and reusable artifacts. The end goal of most software design projects is to deliver software, in our case the UI component, in an effective and efficient manner. Most designers cannot engage in extensive user modeling, and novice designers often have limited expertise in usability engineering. Our method has a clear starting point, and indicates how to proceed from that point to the next phase.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.4. Summary of Proposed Design Method

In this chapter we used the knowledge and expertise elicited from our first case study (see chapter 3) to propose a UI design method with personas and patterns as the primary design directives. We described the key phases of our method: Persona creation, Pattern selection, and Pattern composition. We described a set of involved activities, including a description of lower-level tasks. We note that the list of activities has been compiled based on our design experiences with the NCBI site. Furthermore, we outlined a set of key principles that are the pillars for our design method. These principles aim to balance the UCD philosophy within a more "engineering-like" paradigm. In the subsequent chapters, we will demonstrate how these key principles are deeply entwined with the method, its activities and detailed process.

Recall that the main motivation for our research work is to define a systematic and rigorous *process* for UI Design, with clearly defined steps and tool support. Unfortunately in the current state of the art, our two key artifacts of personas and patterns are defined in purely narrative form and hence not applicable for any form of formal reasoning or sophisticated tool support. Therefore, in the next chapter we propose more formal representations for personas and patterns. Based on these representations we draw semantic relations between both artifacts by defining a set of rules, which are then used in the *UX-P Process* (User Experiences to Patterns Process).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 5

Personas and Patterns: Process Requirements

In this chapter we describe the necessary representational and knowledge requirements for our proposed *UX-P Process*. The actual process steps will be detailed in the subsequent chapter. First, we propose a structure and representation for personas and patterns, to specify the most important aspects of these artifacts. Without a well-defined structure, their practical application within a systematic process is limited. Secondly, we discuss related associations, represented as a set of rules, which draw dependencies between these artifacts.

5.1. Overview

Representations for personas and patterns are necessary to adequately specify them in a format which includes formalizations for machine processing. Current narrative text formats are difficult to manipulate and use for machine processing, while formal descriptions can be used as input for semi-automation. Furthermore, pattern descriptions lack essential information as part of their attribute list, requiring additional information. Therefore, as indicated in Figure 5-1 by (1) and (2), we define a persona and a pattern model. Each model provides a structure for the information and restricts the contents for some variables, to be defined in the sections to follow. In essence, the models can be viewed as representational templates which can be populated with relevant data, resulting in a persona or pattern. Furthermore, we require a set of heuristics and criteria to first specify how to group users into personas, denoted by (3); and secondly, to specify how to select patterns based on a set of personas, denoted by (4). Note that prerequisites for these rules are requirements (1) and (2) since they provide us with the needed formalization.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

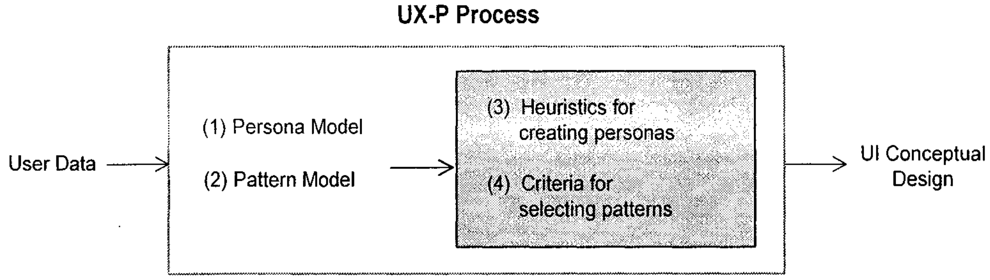


Figure 5-1: Requirements for UX-P Process

In order to fulfill these process requirements, we used a variety of elicitation techniques to gather expert knowledge. In general, knowledge elicitation is the process of acquiring knowledge from domain experts describing either their knowledge about the domain, how they achieve a specific task or procedure, and/or their problem-solving strategy. Acquiring this knowledge from a person and transferring it to a computer program is one use of knowledge elicitation [McGraw and Harbison-Briggs 1989]. We used three different knowledge elicitation techniques, characterized by interaction type: Document analysis, expert focus groups, and semi-structured interviews. Table 5-1 illustrates the techniques and their description.

Technique	Description
Document analysis	Examination of documents and sources that experts use during user analysis and design. Main focus on sources related to creating representative users and designing with patterns.
Expert focus group	Conducted with five experts (both HCI specialists and UI designers) from both academia and industry, and two monitors. Experts were given activities to elicit their knowledge about user attributes and relationships between user variables, usability principles, and design outcomes.
Semi-structured interviews	Interviews were conduced with four experts to determine knowledge used (easily verbalized knowledge) and procedures followed during user analysis and design. In particular, as related to creating representative users and designing with patterns. Experts were a HCI specialist with formal training in cognitive psychology, and three UI designers – with specialties in user modeling, usability factors and metrics, and software engineering.

Table 5-1: Knowledge Elicitation Techniques Used

We then synthesized the elicited knowledge with our own HCI expertise. Table 5-2 illustrates each requirement and cross-references it with the techniques used to derive information.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

#	Requirement	Elicitation Technique		
		Document Analysis	Expert Focus Group	Semi-structured Interviews
1	Persona model	X	X	X
2	Pattern model	X		X
3	Heuristics for creating personas	X		X
4	Criteria for selecting patterns	X	X	X

Table 5-2: UX-P Process Requirements and Elicitation Techniques Used

In what follows, we will detail each requirement and associated outcome.

5.2. Persona Model

Persona descriptions are informal and not suitable for machine processing. We therefore needed to extend these descriptions to include formalizations which are machine readable (input for tools) while keeping their intuitive narrative nature (input for designers). As such, we created a persona model (see Table 5-3) which contains elements for capturing: (1) Textual descriptions for designers, and (2) Variables for machine processing.

Textual descriptions	User variables
• General profile • Goals • Scenarios • Features • Interaction details	• Demographics: Age, Gender, Income level • Knowledge and Experience: Computer experience, Domain experience, Education level, Linguistic ability, Literacy, Product experience • Psychological profile and Needs: Behavior to features (feature keen/shy), Control, Guidance, Initiative taking, Learning speed, Learning style, Learning support, Validation of decisions • Attitude and Motivation: IT attitude, Level of motivation • Special Needs: Disabilities, Special Groups

Table 5-3: Persona Elements

Our starting point was gathering knowledge about how to best represent persona descriptions in a machine-readable manner. We needed to represent a subset of the intrinsically narrative descriptions into information which was both quantifiable and discrete. For example, as binary or integer values representing age and sex of the persona. We elected to use a set of **user variables** to represent the most important user attributes for our personas. We used a number of sources from various domains including HCI, psychology, marketing, and population statistics [Kirakowski and Corbett 1993; Aaker 2004; Statscanada 2006] to first compile, then quantify and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

discretize these attributes.

Our results were first iterated in a formal focus group meeting with five experts who provided information on common practice, and then with a cognitive psychologist (during an interview) who gave further insight on theoretical soundness; we refined and added to the set, removed duplicate variables, and divided specific attributes into more than one variable.

Our goal was to assemble a set of variables which could be extended in the future, but large enough to be used as is in our proposed process. The user variables were grouped into five different categories: (1) Demographics, (2) Knowledge and Experience, (3) Psychological Profile and Needs, (4) Attitude and Motivation, and (5) Special Needs. Table 5-4 illustrates the detailed set of user variables. Each variable is described by name, textual description, and range of values. Some variables had discrete values (belonging to a specific category, such as male and female for age), or were based on a ranking scale (such as none to expert for computer experience). In Table 5-4, unless otherwise indicated, values are based on a ranking scale. We used a semantic-differential ranking scale [Aaker 2004] where the range is based on a five-point scale. The midpoint is neutral. As an example, computer experience is ranked on a scale from 0-4 (none, basic, average, advanced, expert).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

User Variable	Description	Range of Values
1. Demographics		
Age	Age group or range	7 categories: toddlers, children, adolescents, young adults, mature adults, seniors,elderly[10]
Gender	Gender	2 categories: male or female
Income level	Family income, where low income defined as < 50% of median[11]	low to high
2. Knowledge and Experience		
Computer experience	Where basic experience is working knowledge of office systems	none to expert
Domain Experience	Experience in technical or business function supported by product	none to expert
Education level	Formal training and education	7 categories: none, elementary, highschool, vocational, college, undergrad, advanced
Linguistic ability	Knowledge of product language	none to fluent
Literacy	Ability to read, write, use numbers, and handle obtained information	illiterate to fully-functional literacy
Product Experience	Experience with product or with similar software products	none to expert
3. Psychological Profile and Needs		
Behavior to features	Behavior and interaction style towards software features	feature shy to feature keen
Control	Amount of control user needs when interacting with the product	low to high
Guidance	Amount of guidance required when interacting with the product[12]	low to high
Initiative taking	Initiative taking habits of user when interacting with the product	reactive to proactive
Learning speed	Rate this user's learning abilities in general (slow learner/fast learner)	low to high
Learning style	Primary learning style of user	3 categories: auditory, visual, kinesthetic
Learning support	Learning support required when interacting with the product	low to high
Validation of decisions	Validation of decisions required when interacting with the product	low to high
4. Attitude and Motivation		
IT attitude	Attitude to information technology in general	negative to positive
Motivation level	Motivation to use the system	low to high
5. Special Needs		
Disabilities[13]	Physical or intellectual disabilities	Vision (colorblind, low vision, none), hearing, physical/motor, learning/cognitive
Special Groups	Belonging to a special user group	Children, seniors, novice, expert, low literacy

Table 5-4: User Variables in the Persona Model

[10] toddlers (0-4 yrs), children (5-14), adolescents (15-19), young adults (20-34), mature adults (35-59) seniors (60-74), elderly people (75 and over)

[11] middle point on scale defined as the national median, and high income as per bracket for a particular country (i.e. in Canada, as greater than $100 000)

[12] or with similar products

[13] users can belong to more than one group

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

XML (eXtensible Markup Language) was used to represent the model. XML is a commonly-accepted exchange format for the representation of data, and fit our purposes well. The resulting persona model contains both elements for capturing the original narrative descriptions for human processing and reference, along with a set of associated user variables for machine processing. A fragment of a populated persona model is depicted below:

```
<id_key>1</id_key>
<Name> Donna Smith <Name>
<General Profile>
        She is a 24 year-old Masters student in Biochemistry. She lives with a roommate away
        from home. She is quite active and tries to jog daily and play soccer twice a week. She
        uses the internet daily for email access, and searches for biological information related to
        her research... [6 lines omitted].
</General Profile>
<Knowledge_and_Experience>
        <Education Level> 6 </Education Level>
        <Domain Experience> 3 </Domain Experience>
        <Literacy> 4 </Literacy>
        <Computer Experience> 1 </Computer Experience>
        <Product Experience> 0 </Product Experience>
        <Linguistic Ability> 2 </Linguistic Ability>
</Knowledge_and_Experience>

...
```

5.3. Pattern Model

We analyzed the most popular sources that designers exploit for pattern information [Tidwell 2002; Landay 2001; Welie 2003], interviewed three UI designers who have expertise building designs with patterns, and used our own experience with the NCBI study to create a model for representing patterns that is amenable for machine processing.

In current practice, HCI patterns are represented according to the Alexandrian format of "context, problem, solution" [Alexander 1979]. Additional attributes that are typically included are forces, examples and related patterns. Designers use these attributes to assess the applicability of a pattern for a particular design, weighing design trade-offs including usability implications. However, current attributes do not include explicit information about users or usability criteria. Designers often infer this information from other attributes, or make an informed judgment based

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

on their undocumented experiences. The same issue arises when designers want to combine several patterns into a comprehensive design. They have to infer information about the design criteria being addressed by the pattern (such as logical organization or navigation), and the types of relationships with other patterns.

Therefore, to adequately support our process, we extended traditional pattern descriptions in two ways. First, we included supplementary attributes which designers need for both the pattern selection and pattern composition phases: A *short description* of the pattern, which includes keywords from the context, problem and solution; and *relationships* which detail the types of relationships that this pattern has with its related patterns. Secondly, we associated each pattern with a set of **pattern variables** (called P-variables) which would be suitable for machine processing. This set of P-variables has the following information: (1) Primary criteria, which is the main design principle that the pattern addresses, (2) Secondary criteria, the secondary design principle that the pattern addresses, (3) Pattern type, which is the type of library this pattern belongs to, typically organized by domain, and (4) Special needs, which refer to any special user needs that this pattern addresses. The values of each P-variable belong to a discrete and finite domain (see Table 5-5).

Pattern variables	Values
Criteria	Shortcuts/accelerators, Feedback, Error Prevention, Error Handling, Grouping & Structure, Navigation, Consistency, Minimalist Design
Pattern type	Web, GUI, mobile, visualization
Special needs	Colorblind, low vision, no vision, hearing disability, physical/motor disability, learning/cognitive disability, children, seniors, novice, expert, low literacy

Table 5-5: Pattern Variables

As a result, similar to our persona model, we defined a pattern model (see Table 5-6) which consists of both textual descriptions and P-variables; textual descriptions for designers and P-variables for both designers and machine processing.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Textual descriptions	Pattern variables
• Context • Problem • Forces • Solution • Examples • Related patterns • Short description • Relationships	• Primary criteria • Secondary criteria • Pattern type • Special needs (users)

Table 5-6: Pattern model elements

We used XML to represent our pattern model, and included a header tag for the P-variables, and a content tag for the textual descriptions. A fragment of a visualization pattern is illustrated below as an example:

```
<id_key> 25 </id_key>
<Title> Redundant Encoding </Title>

<Header>
    <Primary criteria> Grouping & Structure </Primary criteria>
    <Secondary criteria> Error Prevention </Secondary criteria>
    <Type> Visualization </Type>
    <Special Needs>
        <Group Type> colorblind </Group Type>
    </Special Needs>
</Header>

<Content>
   <Problem>
        How to reinforce the similarities and differences between visual objects.
   </Problem>
   <Solutions>
        Encode the data dimensions using several visual features… [4 lines omitted].
   </Solutions>
   …

</Content>
```

5.4. Heuristics for Creating Personas

Based on the critical examination of the sources that designers use [Cooper and Reimann 2003; Courage and Baxter 2005], interviews with two experts (cognitive psychologist and UI designer with user modeling expertise), and on our own experience with the NCBI study, we propose a set

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of heuristics to create personas. We note that in current practice, HCI experts typically carry out these steps intuitively. Furthermore, existing sources do not explicitly define steps for creating personas, but offer general guidelines.

The heuristics can be divided into two major categories. (1) Heuristics for grouping users based on most important attributes, and (2) heuristics for identifying correlations and dependencies between groups and/or attributes. In particular we propose the following:

- Determine and prioritize the user attributes which are most important with regards to the envisioned design model and context of use.
- Determine the most typical values for each attribute based on users.
- Determine the range and percentage of users falling along the range. Each attribute should be considered independently of the others.
- Consider correlations and dependencies between groups obtained for each attribute.
- When a particular user set seems to repeat itself often across multiple related attributes, it should be considered as a good candidate for a group.
- Each group is used to create a typical user representative, which will eventually result in a persona.

The resulting artifact from these steps is a set of representative users. Each representative user is then supplemented with information such as a name and missing attributes, to result in a persona.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5.5. Criteria for Selecting Patterns

We propose a tentative set of criteria for selecting patterns, with the eventual goal of systematizing this process. These criteria are based on an analysis of the documented design steps and decisions from the NCBI study and knowledge elicitation activities with experts.

We distinguish between three major selection criteria, based on a pattern's: (1) domain of application, (2) direct association with specific user attributes, and (3) inferred association to usability requirements. To illustrate, let us take a simple example like the Wizard pattern (Section 4.2.2). Recall that one of the personas was a novice user who needed guidance and simple navigation. We could have selected this pattern for our design since it is applicable for: (1) web applications, *and* (2) novice users with basic product experience, *and/or* (3) users who need guidance.

The first selection criterion is relatively straightforward since the domain of application for each pattern library is indicated. Furthermore, our proposed pattern model includes this information as a P-variable. The last two selection criteria are more complex, and will be the focus of the remainder of this section.

5.5.1. Documenting the Selection Rationale

We summarize the rationale behind pattern selection in the following:

1. Users vary in their attributes, interaction behaviors, and needs within a specific context. These variations are captured by persona specifications. In our persona model, we describe them with user variables.

2. When not explicitly defined in a persona, user needs are deduced from these persona specifications.
    - For example, a user's need for *efficiency of use* could be deduced from advanced

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

computer experience (user attribute) or by the observation that the user frequently employs shortcuts while interacting with the system.

3. User needs can be divided into (a) usability needs and (b) special needs.
   - An example of a usability need is *efficiency of use*.
   - Specific user groups have special needs. An example of a special need is *avoiding color encoding* for *color-blind* users.

4. Usability needs have inferred implications on physical design, which can lead to pattern selection. These implications on physical design are captured as pattern criteria in our pattern model.
   - For example, *efficiency of use* is a usability need and one physical design implication would be to apply *accelerators*.
   - Among other possibilities, one pattern which is an accelerator is called *macros* [Tidwell 2002].

5. Special needs are addressed within patterns themselves. These needs are specified in our pattern model.
   - For example, the pattern *redundant encoding* indicates in its context that users may have visual deficiencies such as *color blindness* [Wilkens 2003].

6. After gathering the above information, designers can build a mental model understanding the potential pattern, its relationship with other patterns and its applicability in a given context of use. Supplemented by their design experience, they select a subset of these patterns which they compose into a comprehensive design.

As depicted in the above, the decision of whether a pattern is applicable or not is based on a rather complex selection process. As illustrated in Figure 5-2, we can define a set of dependencies associating personas, represented by user variables, to patterns: User variables determine user needs; user needs lead to pattern criteria or patterns; and pattern criteria lead to patterns.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

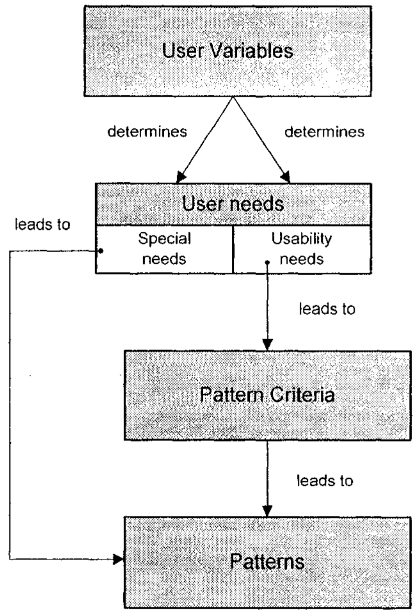


Figure 5-2: Criteria for selecting patterns

Furthermore, we distinguish between the following two user needs since their selection critieria vary:

- **Special Needs**. Persona specifications contain information about whether the represented user group has special needs (contained as a user variable). To systematize associations based on special needs, we can use information that is directly contained in our pattern model. More specifically, the special need is mapped to a P-variable (called "special need") defined in the pattern header of our model.

- **Usability Needs**. Selecting patterns based on usability needs is less straightforward. Designers establish a fuzzy and undescribed association between user attributes, usability needs and patterns. After conducting further knowledge elicitation activities, we created a *usability-design model* (see Figure 5-3) for pattern selection based on usability needs. This model traces in detail the right-hand path of Figure 5-2.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

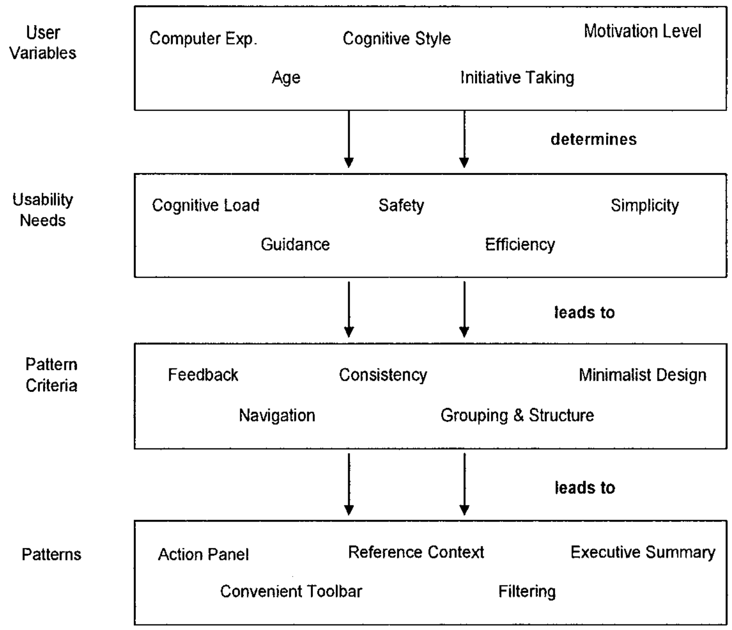


Figure 5-3: Usability-design model for pattern selection

The next few sections will be devoted to explaining how we arrived at the constituents of the above-mentioned usability-design model, in the order of our knowledge elicitation activities: (1) assimilating usability principles, (2) partitioning usability principles into usability needs and pattern criteria, (3) relating usability needs to pattern criteria, and (4) relating user variables to user needs. Finally, we synthesize this knowledge and specify the precise nature of the dependencies; dependencies which will eventually be used for developing pattern selection rules.

5.5.2. Assimilating Usability Principles

A common thread that linked experts together in our interviews, backed by our own experience with the NCBI study, was that user-based design decisions were dictated by usability principles. We studied the four main usability sources that experts use. Each collection of principles is derived heuristically from experience for the design of interactive systems: (1) Bastien and Scapin's Ergonomic Criteria [1993], (2) Nielsen's Usability Heuristics [1991], (3) Shneiderman's Golden Rules of Interface Design [1998], and (4) van Duyne et al.'s Design Principles [2003]. As an example, a subset of Bastien and Scapin's Ergonomic criteria (which consists of 23 criteria

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and sub-criteria) is illustrated in Table 5-7. For the complete set or the other three sources, the reader is referred to the literature.

Ergonomic Criteria	Sub-criteria	Description
Guidance	1.1 Prompting 1.2 Grouping of Items ▪ Grouping by Location ▪ Grouping by Format 1.3 Immediate Feedback 1.4 Legibility	Refers to the means available to advise, orient, inform, instruct, and guide the users throughout their interactions with a computer.
Workload	2.1 Brevity ▪ Concision ▪ Minimal Action 2.2 Information Density	Concerns all interface elements that play a role in the reduction of the users' perceptual or cognitive load, and in the increase of the dialogue efficiency.
Explicit Control	3.1 Explicit User Action 3.2 User Control	Concerns both the system processing of explicit user actions, and the control users have on the processing of their actions by the system.
Error Management	5.1 Error Protection 5.2 Quality of Error Messages 5.3 Error Correction	Refers to the means available to prevent or reduce errors and to recover from them when they occur. Errors are defined in this context as invalid data entry, invalid format for data entry, etc.

Table 5-7: Examples of Bastien and Scapin's [1993] ergonomic criteria

We assimilated all four sources to come up with a key list of usability principles, as illustrated in Table 5-8. The "X" mark along a row indicates the sources which consider a particular usability principle. Although the terminology differed at times (for example, *accelerators* vs. *shortcuts*), all four sources described similar principles. It is noteworthy that Bastien and Scapin [1993] deal with their principles on a more concrete level, and discuss physical design implications and practical applications.

For the purposes of our process, we added three additional principles which we believed were missing from our assimilated list, based on other sources [Padda 1993; Bevan and Macleod 1994; Bevan 1995; Bevan 1997]: **Simplicity**, **appeal**, and **facilitated navigation**. We confirmed our ideas during an interview with an HCI expert who specializes in usability factors and metrics.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Usability Principles	Sources			
	Bastien & Scapin	Nielsen	Shneiderman	van Duyne et al.
Cognitive load (Minimal)	X	X	X	X
Consistency	X	X	X	X
Control	X	X	X	X
Efficiency of use		X		
Error Handling	X	X	X	X
Error Prevention	X	X	X	X
Feedback	X	X	X	X
Logical Organization	X		X	
Guidance	X			
Minimalist Design	X	X		X
Natural mapping	X	X		
Safety		X	X	X
Accelerators		X	X	X

Table 5-8: Assimilated Usability Principles from Various Sources

5.5.3. Partitioning Usability Principles: Needs and Pattern Criteria

Usability principles are cognitive and physical – cognitive because they deal with cognitive factors necessary to create a usable system, and physical because they deal with design factors necessary to create a usable system. We can look at this in the inverse sense, where for the creation of a usable system (in some specific context), a user requires certain cognitive factors fulfilled which are then fulfilled by the associated design factors. For example, *efficiency of use* is a need which is cognitive, while *accelerators* are a physical design implication of satisfying that need. It is also important to note that in this case, we are not dealing with any kind of requirements or needs that are based on performance; e.g. the user may not actually be more efficient in terms of performance, but his/her perception from a cognitive point of view is that he/she is more efficient.

On one hand, principles which deal with cognitive factors are closely related to user variables. Once again, based on our previous knowledge elicitation activities and observations, we knew that designers tended to intuitively establish an association between these principles and user variables. On the other hand, principles dealing with physical design factors were not associated with user variables, but were more closely related to the patterns at the design level.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Usability Need	Simplified Definition
Efficiency of use	Perceived increase in productivity and success in goal completion. Includes the perceived time required to execute a particular task or set of tasks, as well as an increase in the pace of interactions.
Control	Refers to the user's perceived control over system's actions: initiation, execution, processing etc.
Guidance	Refers to perceived means available to guide, advise, orient, inform, and instruct user.
Minimal Cognitive load	Minimize the perceived effort placed on memory and other components of the human body (ex.: hands for typing effort) during interaction.
Natural mapping	Refers to users' immediate understanding of provided information (controls / actions / abbreviations / icons / etc.) based on their cultural standards, real world conventions, known physical analogies and logical interpretation.
Simplicity	It is the property, condition, or quality of being simple or un-combined (not composite). It often denotes perceived purity or clarity. Simple things are usually easier to explain and understand than complicated ones.
Appeal	Refers to aesthetic characteristics; properties of an entity that appeal to the senses. Is usually associated with sensual attractiveness, "goodness" and "beauty".
Safety	State or a feeling implying a low level risk of harm. High levels of perceived safety promote exploration, learning and discovery. Low levels may lead to anxiety in users.

Table 5-9: Usability needs

To account for this difference, we decided to partition our usability principles into two different groups: Usability needs and pattern criteria (Table 5-9 and Table 5-10).

Pattern Criteria	Simplified Definition
Accelerators	Refers to a set of methods, such as shortcuts, which attempt to provide rapid access to certain functionalities and accelerate interaction with the system.
Feedback	Refers to simple, clear and reasonably timed information about actions. Confirmation dialogues, prompting, status information, progress indicators, help, and documentation aids are a few examples.
Error Prevention	Refers to a set of methods which attempt to minimize possible human mistakes during normal usage. Prompting for user confirmation during system-critical tasks and input validation are examples.
Error Handling	Refers to the means deployed to assist the user in recovery from errors. Examples include meaningful error messages such as offering instructions on what went wrong and how to recover in a step-wise fashion, listing an email for technical questions; or better yet, offering to automatically carry out steps that would help them recover from the error.
Logical Organization	Refers to characteristics (such as location, format, behavior, etc.) of objects and actions that should facilitate hierarchical, sequential or simple grouping associations. For example, grouping and distinguishing items that belong together by using color, a specific format, or by positioning them together.
Facilitated Navigation	Refers to the means used to facilitate "movement" through the contents of an interactive program in an intentional manner.
Consistency	Refers to the use of the same design solution when considering a similar context of use during application usage.
Minimalist Design	Refers to the use of only required items in order to create a clean and aesthetic design. Examples include well-designed type, images, and graphical elements.

Table 5-10: Pattern Criteria

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5.5.4. Relating Usability Needs to Pattern Criteria

To relate usability needs to pattern criteria, we first examined all our usability principles sources [Bastien and Scapin 1993; Shneiderman 1998; Nielsen 1994; van Duyne et al. 2003]. Bastien and Scapin's collection provided us with some basic links, but we soon realized we needed to perform a more thorough analysis with experts. We conducted a focus group with five HCI experts. Initially, none of our experts were able to describe specifically what way one entity may affect the other. We asked them to split up into two groups, and to build an E-R diagram to find relationships between the different entities. The results were iterated with a HCI expert who specializes in usability factors and metrics.

Needs \ Criteria	Consis-tency	Error Handling	Error Prevention	Feedback	Logical Org.	Min. Design	Fac. Navigation	Accelerators
Appeal					X	X		
Min. Cognitive Load	X				X	X	X	
Control	X			X			X	
Efficiency of Use					X	X		X
Guidance				X			X	
Natural Mapping	X						X	
Safety		X	X	X				
Simplicity	X					X		

Table 5-11: Dependencies between usability needs and pattern criteria

Table 5-11 illustrates our results. "X" indicates that a dependency exists between a usability need and pattern criteria; i.e. to fulfill a specific user need, designers need to consider the indicated pattern criteria (which in essence act as design criteria). Initial comments from our HCI experts were that they perform this part of the design activity intuitively, and they needed to think back to specific design examples and usability principles before they could clearly identify dependencies. We were quite fortunate since all our HCI experts had experience designing with patterns. We were therefore able to discuss our overall usability-design model, which gave them an idea about the overall picture and how the dependencies would be used.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Saying that however, unlike the dependency between user variables to usability needs (described below), the dependency between usability needs and pattern criteria is undefined. We know a dependency exists, and designers need to consider certain pattern criteria to fulfill a specific usability need, but we do not know how variation in a need affects particular criteria. Later in the chapter, we describe how we overcome this, and still incorporate these dependencies into our process.

5.5.5. Relating User Variables to Usability Needs

The second part of our focus group consisted of asking experts to construct a second E-R diagram to exploit relationships between user variables and usability needs. The results were iterated with a HCI expert who specializes in cognitive psychology. Results are illustrated in Table 5-12. "X" indicates that a dependency exists between a user variable and a usability need.

User Variable	**Usability Needs**							
	Appeal	Min. Cog. Load	Control	Efficiency of Use	Gui-dance	Natural Mapping	Safety	Sim-plicity
Age	X	X			X		X	X
Computer Experience					X	X	X	X
Education level		X			X			X
Initiative Taking			X		X			
Linguistic ability						X		
Literacy		X				X	X	X
Need for Guidance					X			
Validation of Decisions					X		X	
Need for Control			X					
Product Experience			X	X	X	X	X	X
Behavior to Features		X					X	X
Level of Motivation	X							
IT Attitude	X	X			X		X	

Table 5-12: Dependencies between user variables and usability needs

The nature of the dependency is that a change in the user variable will result in some change in the usability need. Although all usability needs are important components of a usable system, there are needs which become more critical for particular types of users. For example, an elderly user's need for minimal cognitive load is greater than a young adult (assuming that the young adult does not have any special needs). Since experts were much clearer on the existing

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

relationships between user variables and needs, we were able to document this knowledge a bit more formally, and quantify most of the relationships. Between a particular user variable (V) and need (N) either an inverse relationship exists, where a decrease in V implies an increase N, or a direct relationship, where an increase in V implies an increase in N. Table 5-13 defines these relationships in detail.

V ↓ implies N ↑			
User Variable (X)	**Values/Ranges**	**Numeric Values**	**Needs affected (Y)**
Education level	none, elem., high school, vocational, college, undergrad, advanced	{0,1,2,3,4,5,6}	SL, CL, G
Linguistic ability	none to fluent	{0,1,2,3,4}	NM
IT Attitude	negative to positive	{0,1,2,3,4}	A, G, CL, S
Initiative taking	reactive to proactive	{0,1,2,3,4}	G
Behavior to features	feature shy to keen	{0,1,2,3,4}	CL, S, SL
Level of motivation	none to high	{0,1,2,3,4}	A
Age	toddler, children	{0,1}*	SL, CL
Age	toddler, children, adolescent	{0,1,2}*	A
Computer Experience	none, basic	{0,1}*	G, S, SL, NM
Literacy	none, basic	{0,1}*	CL, NM, SL, S
Product Experience	none, basic	{0,1}*	G, S, SL, NM
V ↑ implies N ↑			
User Variable (X)	**Values/range**	**Numeric Values**	**Needs affected (Y)**
Guidance	none to high	{0,1,2,3,4}	G
Validation of decisions	none to high	{0,1,2,3,4}	G, S
Control	none to high	{0,1,2,3,4}	C
Initiative taking	reactive to proactive	{0,1,2,3,4}	C
Age	senior, elderly	{5,6}*	SL, G, S, CL
Product Experience	high, expert	{3,4}*	E, C

Legend: Appeal (A); Min. Cognitive Load (CL); Control (C); Efficiency of Use (E); Guidance (G); Natural Mapping (NM); Safety (S); Simplicity (SL)
* Affected needs increase rapidly

Table 5-13: Types of relationships identified for a user variable V and a usability need N

A change in a particular variable may or may not result in a *linear* change in its corresponding need; this is especially true at the extremes. Some needs (denoted by * in the table) changed much more rapidly than others in response to variable changes. Taking both the trend and the rate of change into account reveals four types of relationships between a user variable and a need: (1) a direct relationship that is linear-like[14], (2) an inverse relationship that is linear-like, (3) a direct

[14] note that we cannot say it is precisely linear

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

relationship that is exponential-like[15], and (4) a direct relationship that is exponential-like.

5.5.6. Summarizing the Dependencies

In this section we summarize our findings and observations by presenting the process of pattern selection as a composition of the various dependency relations that exist between the involved selection components, namely user variables, user needs, pattern criteria and patterns. As portrayed in Figure 5-4 we identified five dependencies. The numbers in the diagram correspond to the following:

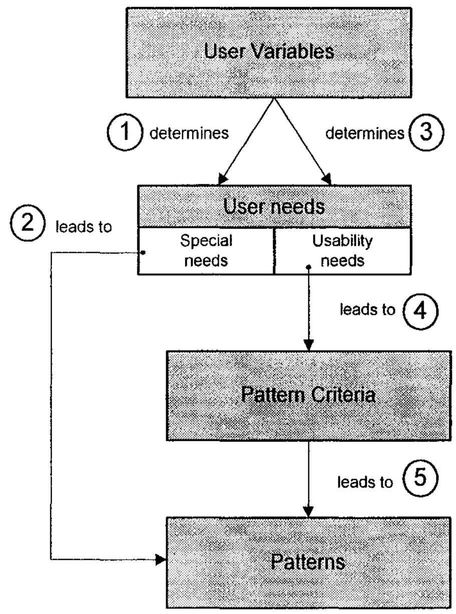


Figure 5-4: Dependencies between selection components

(1) User variables determine special needs. We extracted the following knowledge from experts:

- The binary relation specifying which user variable determines which special need. An example is that the presence of a cognitive disability determines the user's special need for cognitive disability support.
- For each pair of the relation, we determined how a particular user variable relates to a

[15] note that we cannot say it is precisely exponential

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

special need by specifying the extremes of the variation. An example is that users with a cognitive disability require cognitive disability support and users without a cognitive disability require none.

(2) Special needs lead to patterns. We extracted the following knowledge:

- The binary relation specifying which special need leads to which patterns. An example is that a need for cognitive disability support leads to the Simplified Text, Alternative Orthography, and Audio Communication patterns.

(3) User variables determine significant usability needs. We extracted the following knowledge:

- The binary relation specifying which user variables determine which usability needs. An example is that age, IT attitude, and level of motivation determine the user's need for appeal.
- For each pair of the relation, we determined how a particular user variable relates to a usability need by: (a) specifying the extremes of the variation. An example is that novice users require the most guidance and expert users require the least guidance. (b) specifying the trend of the variation. An example is that as the user's computer experience decreases, the need for guidance becomes greater. A more detailed summary of the trends are indicated in Table 5-13.

(4) Usability needs lead to applicable pattern criteria. We extracted the following knowledge:

- The binary relation specifying which usability needs relate to which pattern criterion. An example is that a need for guidance and a need for safety lead to the feedback pattern criteria.
- For each pair of the relation, we determined how a particular usability need relates to a pattern criterion by specifying the trend of the variation. An example is that as the need for simplicity increases, the relevance of minimalist design increases.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(5) Pattern criteria lead to applicable patterns. We extracted the following knowledge:

- The binary relation specifying which pattern criterion leads to which patterns. An example is that the Error Prevention pattern criterion leads to following patterns: Row striping, Multi-level help, Input Hints, Input Prompt, Same-page error messages, Smart Selection, Magnetism, Guides, and On Fly description.
- The relation relates a pattern criterion to all patterns that have the pattern criterion either as their primary criterion or as the secondary criterion (stated in the header section of the pattern). Each pair of the relation is further qualified as either highly applicable (if stated as primary criterion) or applicable (if stated as secondary criterion).

Based on these dependencies, we developed a set of rules to encode the expert's reasoning and logic behind pattern selection. This will be defined in greater detail in the next chapter, on our process.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5.6. Summary of Process Requirements

In this chapter we described the necessary representational and knowledge requirements for our UX-P Process. These consist of more formal representations of personas and patterns, heuristics for creating personas, and criteria for selecting patterns. Requirement outcomes were satisfied by using our experiences with the NCBI study (case study 1) and expert knowledge. The latter was elicited using three different techniques: Document analysis, expert focus group, and semi-structured interviews.

First, we propose a persona model and a pattern model. Each model describes a format for structuring information and restricts the content of the respective artifact to a set of elements. Elements consist of textual descriptions for human processing and variables for machine processing. These models can be viewed as representational templates which are populated with data to describe either a given persona or a pattern.

Secondly, we propose a set of heuristics for creating personas. In current practice, HCI experts typically carry out these steps intuitively. Our heuristics provide the designer with pointers and hints for systematically grouping users based on most important attributes, and for identifying correlations and dependencies between groups and/or attributes.

Thirdly, we documented a set of criteria upon which patterns are selected from a given set of persona specifications. We defined a set of intermediate steps which help in bridging the gap between user variables captured in personas and applicable patterns. In particular, we defined dependency relations between: (1) user variables and special needs; special needs and patterns. (2) user variables and usability needs; usability needs and pattern criteria; pattern criteria and patterns.

In the next chapter, we will use these results to define our proposed UX-P process.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 6

The UX-P Process

Any method is as good as the process supporting it; we therefore developed a systematic process to support our design method. Our process is a UI design process, with the final goal of building a UI conceptual design, based on user experiences captured as personas. In general, a process is an organized set of activities, which transforms inputs to outputs. Specifically, a design process involves human creativity, interactions between a wide range of different people, sound judgment, and background knowledge and experience. In our case, UI designers, HCI experts, and even behavioral scientists can be involved in the process. Finally, a process description should encapsulate knowledge which allows the process to be reused – which is the precise goal of this chapter.

In what follows we propose the User Experiences to Patterns (UX-P) Process and detail its constituting steps. It should be noted that the majority of this chapter takes a white-box approach to describing our process. Our elicitation and documentation activities described previously lay the groundwork for the process description.

6.1. Process Overview

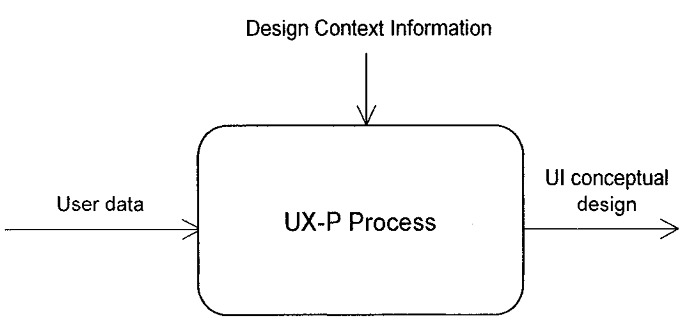


Figure 6-1: Black Box View of the UX-P Process

If we think of our process as a black box, it has two inputs and one output as illustrated in Figure

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6-1. The inputs are: (1) user data, and (2) design context information. The output is a UI conceptual design.

Our process assumes that user data follows the format described in chapter 5. First, we reuse the persona model as a representation for each user. Not all elements from the model are required; the designer may choose a subset according to the design context. Furthermore, textual descriptions for each user are not necessary for the clustering step, but will be an added value for overall persona creation. Depending on time constraints, the designer may choose to include textual descriptions for all or only some of the most representative users. In addition, user variables and textual descriptions not described in the persona model may be added. Values for variables should belong to a discrete domain. Inherently continuous domains should be defined in terms of a small set of ranges, similar to the way we have designated "age" in our set of user variables. Secondly, the design context information plays an important role throughout the process. This information is characterized by results from empirical studies and input from other UCD artifacts. They are used throughout the process to aid designers in making decisions during all three phases.

Figure 6-2 illustrates a process diagram, by depicting the flow of activities involved. As already suggested by our method, the activities themselves are grouped into three distinct phases: Persona creation, Pattern selection and Pattern composition. Note that an overview of our process has been published in [Javahery et al. 2006][16], and a description of the process in pseudo-code is described in Appendix I.

[16] It should be noted that this overview also includes information about extending our process with task variables, which is discussed in the future work of this thesis.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

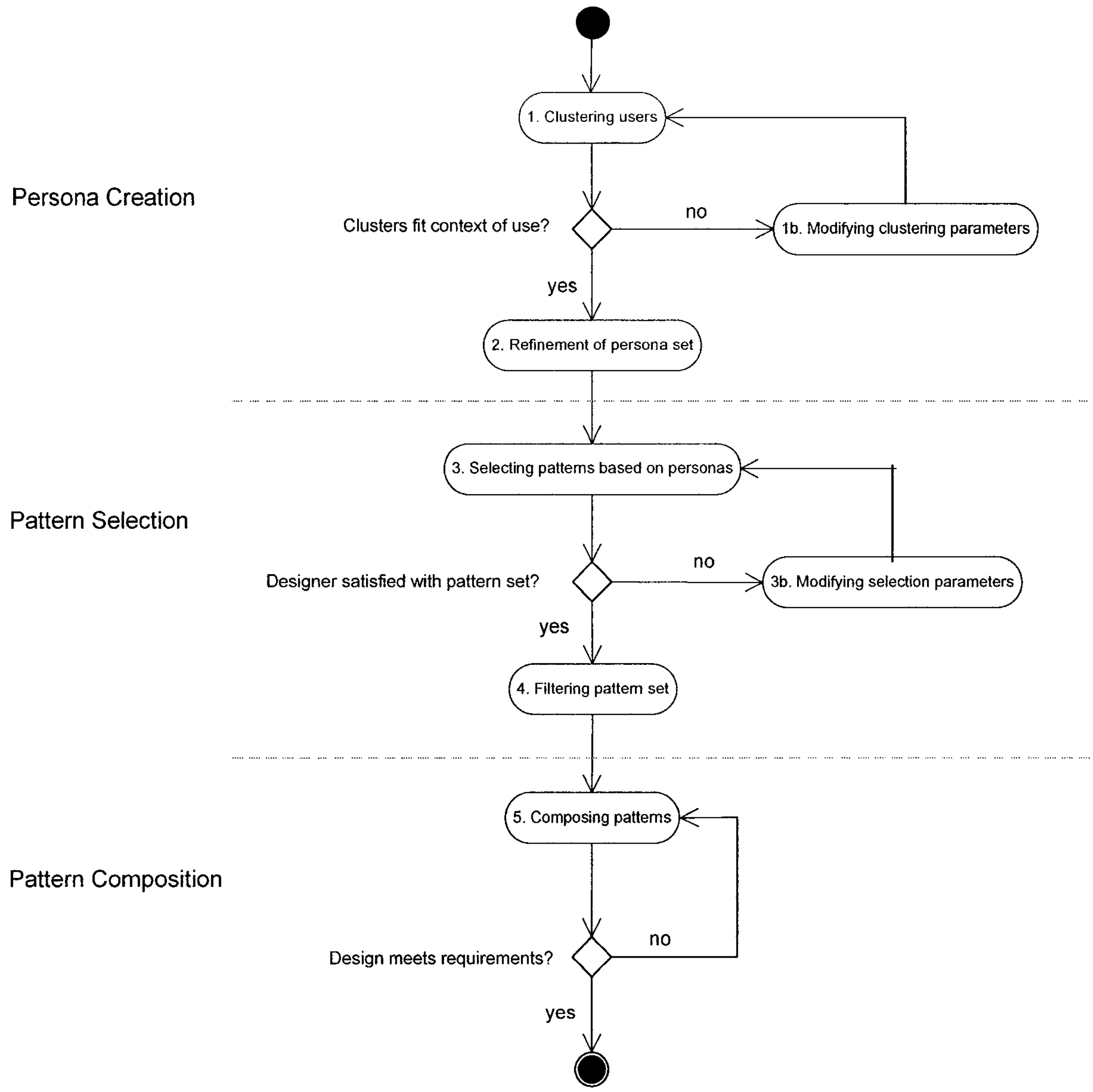


Figure 6-2: Process Diagram for UX-P Process

The persona creation phase is composed of three steps and one decision point. The steps are clustering users, modifying the clustering parameters, and the refinement of personas. The clustering step takes user data, which is the main input into the process, resulting in a set of user clusters. At this point, the designer needs to make a decision about the created user clusters – whether they are adequate and meet the context of use, or if they need to be re-grouped. The designer can then modify the clustering parameters and iterate as many times as required until the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

desired groupings are obtained, and the personas start to take shape. Finally, the designer can refine these personas by modifying or including any additional information.

The pattern selection phase is composed of three steps and one decision point. This phase resides at the core of our process. The steps are selecting patterns based on personas, modifying the selection parameters, and filtering the pattern set. For selecting patterns based on personas, the personas created in the previous step are taken as input. For each persona, patterns are suggested and prioritized, based on a set of rules. The designer has a decision point at this step; if she/he is not satisfied with the pattern set, the selection parameters can be modified. These two steps are repeated until the pattern set is satisfactory. Finally, the designer can filter the pattern set based on the envisioned design model.

The pattern composition phase is composed of one step and decision point. Designers compose patterns by using a pattern-oriented design model and exploiting relationships between patterns. This is purely a design activity, and other artifacts such as task and interaction models may be used. The designer can iterate through various compositions until a suitable pattern-oriented design is attained.

In the following sections, we depict each step in greater detail. We will use supporting examples for each step, which are simple enough to illustrate our point but which capture the essence of each phase. In chapter 7, we will illustrate our process with a single example describing its application as part of a Bioinformatics visualization design project.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6.2. Persona Creation

In this section we describe the steps and sub-steps involved in persona creation.

6.2.1. Clustering users [Step 1 in Fig. 6-2]

To cluster users, the following sub-steps should be followed:

1. Select influential user variables based on variations in interaction behavior and needs, within the particular design context.
2. Prioritize these user variables, and choose a small subset (no more than five) from the top of the list.
3. For each user variable, reduce the set of possible values, to a *restricted domain*. The restricted domain is a set of typical values for that variable, which will allow clustering to be more manageable.
4. For the first variable, determine the users which fall under each value. Each set of users results in a cluster.
    a. If the sample size is large, percentage of users can be used.
    b. Users who fall outside these values should be placed according to the designer's knowledge and experience. If the value of the user variable is deemed to be important in terms of the design context, the restricted domain can be refined so as not to lose this information.
5. Dependencies between variables should be considered. A dependency results when a combination of variables causes some interaction behavior, but this behavior does not result when each variable is considered in isolation.
    a. This may result in multiple variables being considered together for clustering.
    b. For example, if the combination of age group and IT acceptance causes a significant behavioral change, we need to preserve this information (see example below).
6. Clusters need to be analyzed by considering their importance to the design context. Clusters which are not representative of the design context should be removed. Affinity diagrams can be used if the clusters are too numerous to handle.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

a. However, note that the percentage of users per value only serve as a guide to creating clusters. For example, only 3% of users from a sample may fall under the category of "child". This may be due to sampling constraints with younger users and not be representative of the target group. Furthermore, recall that personas do consider boundary cases which may be important for the design context.

7. For the next user variable, take the set of clusters and break them into sub-categories following steps 3-6. Repeat until you have created clusters based on all user variables in your selected list.
8. For each cluster, construct a representative user. Use the most representative values for each variable. This new user will serve as the basis for a persona.

Let us demonstrate the above sub-steps with a hypothetical example. We will carry this example throughout our persona creation phase. Note that references made to usability studies are illustrations used to exemplify our point[17]. To set the context of use, the application we will be designing for is a mobile phone game called *shop 'til you drop*. In this game, users go to virtual rooms and with every shopping purchase they receive points. Purchases include items categorized as clothing and fashion, household items, or electronics and gadgets. The goal is to be a "smart" shopper and attain a certain amount of points within a specific timeframe. Frequent players with high scores receive shopping incentives and gifts from sponsors.

Before clustering, let us assume that we conduct an appropriate user and domain analysis, where we gather information from potential users of the application. We have access to market segmentation data for mobile phone use, but not specific information relative to the application since it is a new concept. We conduct usability studies with 20 potential users by administering questionnaires and observing interaction behavior with a low-fidelity prototype. The users are current mobile phone users with subscriptions that include games. We represent each user in our sample with the most relevant variables from the persona model. One user, Chantal, is illustrated in Table 6-1. We have added one variable not present in the persona model, but which we believe may influence interaction behavior – *IT acceptance*. Note that IT attitude is described in the

[17] They are not based on actual usability studies which we carried out, but are used for illustrative purposes.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

model, but not IT acceptance. The variable is added with these possible values: {early adopter, average adopter, technophobe}. We also decide to include a textual description of the general profile and interaction details.

User Variable	Values
General Profile	Chantal is a 17 year-old high school student, in her final year. She loves to find out what the latest trends are in fashion. She has a social life, although not as active as she would like, since she just moved to a new city. She uses her mobile phone to play games when she is bored.
Interaction details	Chantal gets easily frustrated if system takes too long to respond. She likes to explore first before selecting an item to purchase[18]. She engages in short playing times, but frequently (daily). She engages in shopping sprees with her friends.
Age	Adolescent, 15-19 yrs old (2)
Gender	Female
IT acceptance	Early adopter
Income level	Student, somewhat low (1)
Computer experience	Average (2)
Education level	High school (2)
Linguistic ability	Somewhat fluent in English (3)
Product Experience	Advanced, with similar products (3)
Behavior to features	Average (2)
Control	Somewhat high (3)
Guidance	None to low (0)
Learning speed	Somewhat high (3)
Validation of decisions	Somewhat low (1)
IT attitude	Somewhat positive (3)
Level of motivation	Somewhat high (3)

Table 6-1: Representation of the user Chantal with persona model elements

We analyze the gathered evidence. Based on variations in interaction behavior and needs, we select our first set of influential variables in order of priority: *Age* and *IT acceptance*. If pertinent, we now need to reduce the possible values for each variable to a restricted domain. Considering that the value for age has seven different possibilities, and since all values are not applicable for our design context, we reduce the domain. In particular, we omit toddlers, children, and the elderly since they are infrequent mobile phone users.

[18] Since application is a new concept, this was gathered during testing with a low fidelity prototype.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

User variable	Restricted domain	Percentage of users
Age	Adolescent	28
	Young adult	37
	Mature adult	27
	Senior	8
IT acceptance	Early adopter	35
	Average adopter	55
	Technophobe	10

Table 6-2: Current mobile phone users with gaming subscriptions

We determine the percentage of users for each user variable (Table 6-2) and restricted domain. Recall that our sample is based on mobile phone users with gaming subscriptions, which we believe could be potential users of our application. Right away, seniors and mature adults are removed since they are not part of the target users of the application. Therefore, the age variable results in two clusters: *Adolescent* and *Young adult*. The two resulting clusters are then sub-categorized based on IT acceptance. We exclude technophobes from the possibilities for IT attitude; based on the user studies which demonstrate that these individuals rarely use mobile phone games (although 10% of them do use a mobile phone). As illustrated in Table 6-3, there are four resulting clusters (C1, C2, C3, C4).

User variable 1: Age Group	User variable 2: IT acceptance	Clusters	Decision
Adolescents	Early adopter	C1	keep
	Average adopter	C2	remove
Young adults	Early adopter	C3	remove
	Average adopter	C4	keep

Table 6-3: User clusters

Evidence from the usability studies indicate a major difference in shopping interaction behavior between adolescents and young adults, where adolescents engage in more "shopping sprees" and young adults are more careful with their spending habits. Therefore, it was important to capture this difference. Studies also indicate differences, although subtle, in interaction behavior between early adopters and average adopters; where early adopters were more likely to explore the available entertainment options on their mobile phone, while average adopters need some

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

guidance or social incentive for the initial gaming experience (i.e. encouragement from peers). Since no dependency was found between the two age groups (adolescent and young adult) and their IT acceptance (early and average adopter), two clusters were omitted: C2 and C3.

The key point here was to preserve information causing differences in interaction behavior; C2 and C3 can both be captured by the combination of C1 and C4. In particular, by designing for C1 and C4, we can also satisfy the needs of C2 and C3. Note that we could have also easily done the inverse and chosen C1 and C4. However, based on the real user sample, we determined that representative users were found more in C1 (adolescent and early adopter) and C4 (young adult and average adopter). In other words, based on the real user sample, more of the sample was adolescent and early adopter vs. adolescent and average adopter; similarly, for young adult and average adopter.

Therefore, our resulting clusters are (1) Adolescents and early adopters, and (2) Young adults and average adopters.

6.2.2. Modifying clustering parameters [Step 1b in Fig. 6-2]

The sub-steps from the previous section describe the most common path for clustering. However, an alternative path may occur after sub-step 7, where the influential user variables need to be re-selected, and the designer has to re-iterate starting from the first sub-step. This may happen after further usability studies have been performed, where designers realize that they have not clustered according to the correct user variables or values, and that they have omitted important parameters.

Let us illustrate this and continue with our *shop'til you drop* example. We may consider a third variable: *Gender*. Let us assume that further analysis of the usability studies with the low-fidelity prototype indicate differences in needs and interaction behaviors between females and males. Two examples being: (1) Females showed a greater level of interest in playing the game as compared to males, and (2) Females spent time browsing most of the virtual rooms, whereas males spent most of their time searching for electronics and gadgets. Therefore, we add gender as

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

a third variable to result in eight possible clusters (C1 to C8). This is illustrated in Table 6-4.

User variable 1: Age Group	User variable 2: IT Acceptance	User variable 3: Gender	Clusters	Decision
Adolescent	Early adopter	female	C1	keep
		male	C2	keep
	Average adopter	female	C3	remove
		male	C4	remove
Young adult	Early adopter	female	C5	remove
		male	C6	remove
	Average adopter	female	C7	keep
		male	C8	remove

Table 6-4: Second iteration of user clusters

We will now attempt to reduce the number of clusters to capture only the most essential information. Following the same reasoning from the previous section, we only need to keep C1, C2, and C7: (C1) Adolescent, early adopter, female, (C2) Adolescent, early adopter, male, and (C7) Young adult, average adopter, female. Information contained within the rest of the clusters can be captured by C1, C2 and C7. Furthermore, other sets are possible, but we chose this particular one since it follows closely with the representative users. As an example, more males were early adopters and adolescents for this type of application. Therefore, this male representative was chosen.

To further explain our decisions about the user clusters, recall first that differences in interaction behavior were observed for users in each age group, and similarly for users varying in IT acceptance. However, the combination of age group and IT acceptance did not result in any behavioral differences. Therefore, we have to simply preserve each value without being concerned about any combinations: *Adolescent*, *young adult*, *early adopter*, *average adopter*. When we add gender, the scenario changes slightly. We need to capture this information, similarly to the other two variables since behavioral differences were observed between males and females. In addition, we note that there is a dependency between gender and age group for females. In particular, females that are young adults spend more time purchasing household items, whereas females that are adolescents spend more time purchasing clothing and fashion

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

items. We decide that this variation is important for our design context, and the resulting clusters need to therefore preserve this particular information: *Female adolescents*, and *female young adults*.

We decide that further iterations are not necessary. Therefore, the final clusters are: (1) adolescent, early adopter, female, (2) adolescent, early adopter, male, and (3) young adult, average adopter, female.

6.2.3. Refinement of Persona Set [Step 2 in Fig. 6-2]

The clusters act as a skeleton for creating personas. To manage and group user information, we clustered based on a small number of influential variables. However, as discussed in previous chapters, personas are artifacts which both designers and developers use for guiding the development process. They are used as a communicative tool. Therefore they need to be engaging, and described in an intuitive and life-like manner. To accomplish this and form concrete personas, it is important to complete information for each cluster by adding textual descriptions and detailed variables. The persona model described in chapter 5 is used for this purpose. The detailed sub-steps are:

1. User variables which were used to create the final set of clusters are called *identifiers*. Use these identifiers to first build a skeleton of each persona.
2. Add an identity (name) and initial general profile for each skeleton. This will be helpful in guiding the creation of the rest of the persona. If available, you may use representative profiles from original user descriptions.
3. For each persona skeleton, populate the remainder of the user variables and textual descriptions.
    a. Follow the persona model as a guide.
    b. Use representative values and information from original user descriptions.
    c. Be aware of certain interaction behaviors characteristic of a persona which may not have been used for clustering, but which will have an impact on user variables and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

textual descriptions. These are called *hidden dependencies*, and it is important to capture them in the personas since they may have an impact on pattern selection. An example of this will be illustrated below, where one of the clusters "adolescent, early adopter, female" also has characteristics of an *expert* user; including a need for *efficiency*.

d. For easier readability, user variables can be converted to textual attribute descriptions. This is optional.

4. Add a picture and personal quote for each persona.

To illustrate, we will carry on with our *shop 'til you drop* example. Table 6-5 illustrates the three persona skeletons extended that result from our clusters. Each persona is given an identity (first and last name) and an initial general profile. Note that for this step onwards, we use representative values and information from the original user descriptions as a guide.

Persona 1: Anna Spinelli Identifiers: Adolescent, early adopter, female	**Persona 2: Jason Li** Identifiers: Adolescent, early adopter, male	**Persona 3: Chantal Larose** Identifiers: Young adult, average adopter, female
Anna is an 18 year-old college student. She loves to shop and hang out with her friends. She uses her mobile phone to keep contact numbers, records all important events in the calendar, and likes playing games on it. She likes to try new games often.	Jason is a 16 year-old high school student. He goes to the arcade often after school. His mobile phone is always ringing, and he often plays his favorite games on it. He likes to explore new games only occasionally. He loves gadgets, and uses his spending money to add to his collection.	Chantal is a 29 year-old office assistant. She loves to find out what the latest trends are in fashion. She has a social life, although not as active as she would like. One of her friends told her about *shop 'til you drop* and she likes to compete for the shopping incentives.

Table 6-5: Persona skeletons

As an illustration, we will complete the first persona, whom we have called Anna Spinelli. We populate the rest of the user variables, as illustrated in Table 6-6.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

User Variable	Values
Age	Adolescent, 15-19 yrs old (2)
Gender	Female
IT acceptance	Early adopter
Income level	Student, somewhat low (1)
Computer experience	Average (2)
Education level	College (4)
Linguistic ability	Fluent (4)
Product Experience	Advanced, with similar products (3)
Behavior to features	Average (2)
Control	High (4)
Guidance	Somewhat low (1)
Learning speed	High (4)
Validation of decisions	None to low (0)
IT attitude	Positive (4)
Level of motivation	Somewhat high (3)
Disabilities	None
Special Groups	Expert

Table 6-6: Populating user variables for the persona Anna

We notice one main hidden dependency. Anna, similarly to the users she represents, also has characteristics of an expert user (part of *special needs* in the user model). Furthermore, she has a need for *control* and *efficiency*, but at the same time, her interaction behavior indicates that she also has a need for *exploration*. These hidden dependencies are important to capture and note since they may have an impact on pattern selection.

Finally, we add textual descriptions (complete general profile, goals, a scenario, etc.) as indicated in the user model. We also decide that for easier readability for designers, we will convert the user variables to textual attribute descriptions. To bring our persona to life, we add a picture and personal quote. The narrative view of the final persona[19] is illustrated in Table 6-7.

---

[19] It should be noted that all identities (names and pictures) used for personas throughout this thesis are fictional. Any similarities to actual individuals are purely coincidental.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Persona 1: Anna Spinelli		**"I just want to play for a few minutes when I am bored. I want to beat my friends and be the 'smartest' shopper"**
Identifiers	Adolescent, early adopter, female	
General Profile	Anna is an 18 year-old college student. She lives with her parents. She is in her first year of the Commerce program, and is involved in different extra-curricular activities like intramural soccer and the social club. She has a part-time job working at a local movie theatre. She loves to shop and hang out with her friends. She uses her mobile phone to keep contact numbers, records all important events in the calendar, and likes playing games on it.	
Goals	Professional: Succeed in school, and work towards a marketing university degree. Personal: Enjoy time with her friends and family. Application: Have fun for a few minutes, and then get back to studying.	
Scenario	Description	Anna has a 30 minute break between two of her classes. She is sitting in the college cafeteria with her friend Angy, who is sitting beside her studying. She is bored and doesn't feel like studying anymore, so she puts away her books and goes to the "games" options on her phone. She chooses *shop 'til you drop*. She gets frustrated because it takes so long to load. She really wants to beat her score from last time. She likes to move often from one virtual room to another, and have control over her surroundings. She likes to explore the different shopping items before deciding on an item to purchase. She plays the game for 10 minutes, but the beeping sound bothers Angy and she has to turn it off.
	Specific needs	Control, Efficiency, Exploration
	Features	Quick loading, scoring recall/tracking, rapid-key exploring, silent mode (visual indicators)
	Interaction details	Anna gets easily frustrated if system takes too long to respond. She is very competitive and wants to keep close track of scores. She likes to explore first before selecting an item to purchase. She engages in short playing times, but frequently (daily).
Demographics	She is an 18 year old female, with a student income and financial support from parents. She has a part-time job which allows her to pay for social and personal expenses not related to school.	
Knowledge and Experience	Anna is a college student and a native English speaker. She has average experience with computers and is an advanced mobile phone and game user. She has been playing mobile phone games for 2 years now, but only started to play *shop 'til you drop* a few months ago.	
Psychological profile and needs	She needs to be in control, and is a fast learner (high learning speed). She needs basic (low) guidance and no validation of decisions.	
Attitude and Motivation	She has a positive attitude to IT, and somewhat of a high level of motivation to use the system.	
Special Needs	She has no disabilities but belongs to a special user group, *experts*.	

Table 6-7: Narrative View of the Persona Anna

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6.3. Pattern Selection

In this section we first describe a rule-based scoring technique that we propose for pattern selection. We will then define the three steps of the pattern selection phase: Selecting patterns based on personas, modifying selection parameters, and filtering the pattern set.

6.3.1. Rule-based scoring technique

As presented in section 5.5.6, user variables are associated with patterns through a set of dependencies. We determined the nature of the dependencies and their relative weights based on expert input. To allow for computation, we propose to use these weights as part of a rule-based scoring technique, along with a number of assumptions that will be discussed below. The scoring technique comes from ideas gathered from a recommender system [Kim and Kim 2003], where suggestions are made based on a computation of the confidence of the result i.e., the score. That is, if a persona is described by a set of values, the confidence of a pattern suggestion is a sum of all confidences of the rules that have been used to compute that result. We use a similar approach to [Kim and Kim 2003] where the confidence of each rule is determined empirically, and their summation is used to compute scores, resulting in a final recommendation.

To keep track of the propagation of scores along dependency links, we use a directed graph representation consisting of a set of start nodes (input), end nodes (output) and inner nodes that store temporary results. Furthermore, nodes are typed. The first type of node represents user variables, which are also input nodes. The second and third types of nodes represent user needs and pattern criteria, which are inner nodes. The fourth type of node represents patterns, which are end nodes. Each node is assigned a value. The propagation of the values from the start nodes to the end nodes is determined by (1) the edges linking the nodes, and (2) the weight assigned to each edge traversed. Our scoring technique has been derived by observations made during knowledge elicitation. To be able to quantify the parameters and dependencies involved, we made a number of assumptions. These are discussed as part of the associations below:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- There exists a direct association between user variables indicating a special need, and patterns addressing this need. Accordingly, if a persona has a special need then all patterns addressing this need receive the highest score possible ('max').
- The association between user variables and usability needs can be approximated by the trends and rates of change, summarized in section 5.5.6. Users fall into categories between the extremes of any particular variable, and as a result, we assigned a value to each of these categories. Please see Table 6-8.
- The association between pattern criteria and patterns are as follows: For each pattern, there is a pattern criterion that is *highly* applicable, and a pattern criterion that is applicable. As a result, if the pattern has the pattern criterion as a primary criterion then it inherits all of its scores. If the pattern has the criterion as a secondary criterion then it inherits half of its scores. Examples of primary and secondary criteria for specific patterns are illustrated in Table 6-9.

It is important to note that our knowledge elicitation activities concentrated on establishing trends and rates of change. We knew that the initial numbers we assigned could be fine-tuned later, based on the assessment of the quality of the results, much as is done for weights in neural networks and rule confidences in expert systems. This will be discussed further in the conclusion of this thesis.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Variable	Association to Need	Variable	Association to Need
Linguistic Ability	As linguistic ability decreases NM increases proportionally	Need for guidance	As need for guidance increases G increases proportionally
Literacy	When literacy is 0 (illiterate) S, SL, NM, CL increase by 4 pts When literacy is 1 (low) S, SL, NM, CL increase by 3 pts	Need for validation of decisions	As need for validation of decisions increases G and S increase proportionally
Computer Experience	When computer exp is 0 (none) G, S, SL, NM increase by 4 pts When computer exp is 1 (low) G, S, SL, NM increase by 3 pts	Need for Control	As need for control increases C increases proportionally
Education Level	As education level decreases SL, CL, G increase proportionally	Behavior towards features	As behavior towards features decreases CL, S, SL increases proportionally
Initiative taking	As initiative taking decreases G increases proportionally As initiative taking increases G increases proportionally	IT attitude	As IT attitude decreases A, G, CL, S increase proportionally
Age	When age is 0 (toddler) A increases by 4 pts SL, CL increase by 3 pts When age is 1 (child) A increases by 3 pts SL, CL increases by 2 pts When age is 2 (adolescent) A increases by 2 pts When age is 5 (senior) SL, G, S, CL increase by 3 pts When age is 6 (elderly) SL, G, S, CL increase by 4 pts	Product Experience	When product exp is 0 (none) G, S, SL, NM increase by 4 pts When product exp is 1 (low) G, S, SL, NM increase by 3 pts When product exp is 3 (high) E, C increase by 3 pts When product exp is 4 (expert) E, C increase by 4 pts
		Level of motivation	As level of motivation decreases A increases proportionally

Legend: Appeal (A); Min. Cognitive Load (CL); Control (C); Efficiency in Use (E); Guidance (G); Natural Mapping (NM); Safety (S); Simplicity (SL)

Table 6-8: Rules for associating user variables to usability needs

Pattern Name	Short Description	Primary criteria	Secondary criteria
Action panel	UI panel for related actions instead of menus	Logical Org.	Accelerators
Corner Treatments	Use corner treatments instead of right angles	Appeal	Consistency
Details on Demand	Display item details in a separate window	Minimalist Design	Feedback
Multi-level help	Mixture of help techniques for varying needs	Feedback	Error Prevention
Sequential	Organize pages in a sequence	Navigation	Logical Org.

Table 6-9: Examples of patterns and related pattern criteria

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6.3.2. Selecting Patterns based on Personas [Step 3 in Fig. 6-2]

For this step, an appropriate pattern library needs to be used, with patterns that are described according to the pattern model. Pattern libraries from various domains may be mixed, depending on the design context. For example, both web and visualization patterns are relevant for a web application which allows users to visualize 3-D bioinformatics data.

Each persona will result in a distinct set of candidate patterns. Patterns are selected based on personas as per the following sub-steps:

1. For the first persona, select a set of the most influential variables (and their associated values) for your design.
    a. These variables should be based on the *identifiers* and the discovered *hidden dependencies* from clustering.
    b. Further variables may be chosen based on the persona descriptions.
2. Since our scoring system is dynamic, use a labeled graph representation in order to keep track of the propagation of scores. The edges (and their direction) of the graph are defined based on the dependency relation between user variables and user needs.
3. The start nodes are user variables. Indicate each user variable's initial value as per your persona descriptions.
4. For each edge defined between a node representing a user variable and a node representing a usability need; compute the score according to the rules defined in Table 6-8 (and Appendix F) and assign the edges with the score.
5. Compute the score for each node representing a usability need, which equals to the sum of all edges leading to it.
6. For each edge defined between a node representing a usability need and a node representing a pattern criterion (see Appendix F), assign it the score which is associated with the node representing the usability need.
7. Compute the score for each node representing a pattern criterion, which equals to the sum of all edges leading to it.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

8. For each edge defined between a node representing a pattern criterion and a node representing a pattern, assign the score which is associated with the node representing the pattern criterion.
9. Compute the score for each node representing a pattern by adding the score of the edge coming from its primary pattern criterion plus half of the score of the edge coming from its secondary pattern criterion.
10. For each edge defined between a node representing a special need and a node representing a pattern assign the score of "max".
11. Organize the patterns in a table, ordering based on score with highest scores first ("max" scores from special needs are considered highest).
12. Repeat the above steps for each persona.

Let us demonstrate the above sub-steps with an example. To set the context, we will be designing an easy-access website. As part of a new provincial initiative, the city of Montreal wants to support accessibility to information for the general public by redesigning a portion of its site. Let us assume that our clustering step has produced two different personas for this particular context: (1) Serge Lacroix, a 51-year old man with low literacy, and (2) Sandrine Dupuis, a 39-year old career woman. The needs of the two personas are so different that we would need two versions of the website (i.e. a compromise solution like in the NCBI study from chapter 3 is not feasible). For the sake of our example, we will therefore only concentrate on Serge's version of the website.

Literacy	Low (1)
Validation of decisions	High (4)
Education level	Elementary (1)
Linguistic ability	Average (2)
Computer Experience	Basic (1)
Guidance	Somewhat high (3)
Initiative taking	Somewhat reactive (1)
Learning speed	Somewhat low (1)
IT attitude	Negative (0)
Level of motivation	Average (2)
Disabilities	None
Special Groups	Low literacy

Table 6-10: User variables of the Persona Serge

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

A set of the most important user variables for Serge are illustrated in Table 6-10, and his persona is described in Table 6-11.

Persona 1: Serge Lacroix		**"I want to be able to find information online without having to ask my son for help"**
Identifier	Low literacy	
General Profile	Serge is a 51-year old city worker. He lives with his wife of 30 years. He has three grown children and two grand-children. He is a blue-collar worker, and has been working for the city for over 25 years. He is worried about his retirement, and is careful with his spending habits. He has a decreased ability to read and write, although he is more comfortable with using numbers. For public service inquiries, he usually makes telephone calls. For figuring out banking and billing information, he gets help from his son. Through a new initiative with the city of Montreal, he heard about easy-access information on some websites, which is now supporting users like him. He is nervous about dealing with the internet, but he would prefer this over asking his son all the time for help. For fun, he likes to kick back with some of his friends, enjoy some beer, and watch hockey games.	
Goals	Professional: Maintain his job until retirement. Personal: Increase his savings, spend time with his grand-children. Application: Find basic information.	
Scenario	Description	Serge is home for the evening, and wants to find some information on the internet about the *Montreal Access Card.* His co-worker told him that the card can give him discounts on a number of services in Montreal, including recreational and sports-related entrance fees, and that he can find all the information online. He wants to click on a link which he thinks will lead him to the right page, but he is unsure. He hesitates for a few seconds, and then decides to click. He tries to make sense of the sentences on the page, but there is way too much information. He isn't sure where to click next. He gets tired, and calls it a night, deciding that maybe he will try again tomorrow.
	Specific needs	Validation of decisions, Simple interface
	Features	Less text, more visual indicators, easily-structured information, easy-to-read fonts, static information, simple navigation
	Interaction details	Serge gives up easily if he can't find the information he needs. He is worried about clicking on a link when he isn't entirely sure of the outcome and content. He doesn't trust his own abilities when interacting with the computer. He feels nervous when there is too much information and he can't make his way around a site. He has an easier time making out words in lists and when large fonts are used.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Demographics	Serge is a 51-year old male, with somewhat of a low family income. He has two financial dependents (his wife and mother).
Knowledge and Experience	Serge is a native French speaker. He has a low literacy level, and did not finish elementary school. He has little (basic) experience with computers and the internet.
Psychological profile and needs	He has an extremely high need for validation of decisions when interacting with a computer system. He is somewhat of a reactive user, and has a low learning speed. He also has a somewhat high need for guidance during system interaction.
Attitude and Motivation	He has a negative attitude to IT, but an average level of motivation to use the system.
Special Needs	He has no disabilities but belongs to a special user group, *low literacy*.

Table 6-11: Narrative view of the Persona Serge, a low-literate user

The first step is to select a set of influential variables based on the persona and design context. Based on Serge's interaction behavior, his needs, and the design context, literacy (identifier) and need for validation of decisions (hidden dependency) are selected. Serge has a strong need for validation of decisions (value of 4) and a low literacy level (value of 1). As illustrated in Figure 6-3, a graph representation is used to keep track of the propagation of scores. Note that the white nodes are user variables, light grey are user needs, and dark grey are pattern criteria. Patterns are represented by rectangles. Furthermore, we have simplified the example for illustrative purposes and have not included all the nodes.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

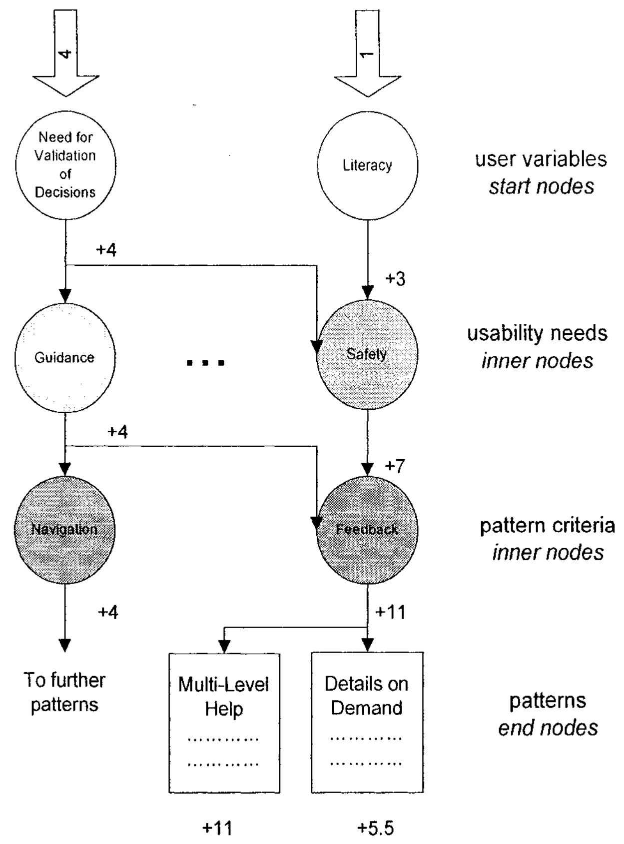


Figure 6-3: Example of rule-based scoring technique

First, user variables (start nodes) are obtained directly from the persona. Initial scores are determined based on the rules in Table 6-8. As a result, the edge between *literacy* and *safety* has a weight of "3" and the edge between need for *validation of decisions* and *safety* has a weight of "4".

To compute the value for the node *safety*, we take the sum, resulting in a value of "7" for safety.

These scores propagate to the related pattern criteria, where each criterion receives the sum of scores of the attached incoming edges. For example, the *feedback* pattern criteria node receives the score equal to the sum of its incoming edges, which in our example is "11". Next the score of the various pattern criteria propagates to the (end) nodes representing the various patterns. End nodes have at most two dependencies. The edge between *feedback* and the *multi-level help* pattern has a weight of "11". Recall that each pattern is associated with two criteria, a primary criterion and a secondary criterion. Feedback is a primary criterion for multi-level help, and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

therefore the pattern gets a full score of "11". However, for *details on demand*, feedback is a secondary criterion and therefore gets half of the score of 11. At this point we note that the score for both the multi-level help pattern and the details on demand pattern are temporary as both patterns will further inherit points from their respective secondary and primary criteria.

#	Title	Short description
1	Simple Text	Organize content using simple and short text
2	Alternative Orthography	Write content phonetically (phonetic spelling)
3	Audio Communication	Provide content in audio format
4	Filter	Filtering facilities to reduce number of data items
5	Stack Page	Use several surfaces stacked together to group content
6	Global Navigation	Consistent set of links to navigate to key sections
7	Sequential	Organize pages in a sequence (linear structure)
8	Hierarchical	Organize pages using a hierarchical cascade model
9	Multi Level Help	Mixture of help techniques to support varying needs
10	Reference Context	Refer graphical objects visually to reference context
11	Wizard	Step user through tasks and maintain visibility
12	Illustrated Choices	Use pictures instead of words to show choices
13	Few Hues, Many Values	Choose at most three color hues for the interface
14	Details on Demand	Uses a separate window to display item details
15	Input Hints	Place an explanation/example beside text field
16	Contrasting Font weights	Separate levels of info by using contrasting font weights
17	Smart Selection	Automatically select coherent group of items
18	Tiled Page	Divide the page into panes so that user can find relevant info
19	Tree Table	Illustrate tree structure of hierarchical data by using columns
20	Overview plus Details	Place overview of information next to zoomed detailed view

Table 6-12: A subset of recommended patterns

The pattern library we chose to use in this context is a combination of GUI [Tidwell 2002], web [Javahery and Seffah 2002; Welie 2003] visualization [Wilkens 2003], and our own special needs patterns. The total number of patterns in the library is 83. A list of the highest recommended patterns is depicted in Table 6-12. The patterns are ordered according to their scores, where patterns are listed in order of descending scores.

Let us now compare Serge to the selected patterns. First, Serge has a low literacy level. From our persona description, we see that he requires a simple interface with a low level of language complexity. He also has a high need for validation of decisions, requiring the system to reassure and confirm his actions.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The top three selected patterns, # 1-3, are special needs patterns and are aimed at addressing the needs of Serge and similar users, by providing a low level of language complexity. More precisely, when used in combination, these patterns provide three alternative communication means assuming different levels of low literacy. For instance, the *Simple Text* pattern assumes that the user is capable of reading and understanding short phrases organized in simple and short text. *Alternative Orthography* assumes that the user can make out linguistic constructs and pronunciation, but does not know how to put letters in the alphabet together. Finally, *Audio Communication* assumes that the user does not know the alphabet and can communicate only orally.

Patterns # 4-8 include *Filter*, *Global Navigation* and *Sequential* patterns. Most of them focus on reducing complexity, grouping structure and organization, which addresses the need of Serge who cannot handle complexity well. Secondly, the Filter pattern is a visualization pattern which aims to reduce the number of visual objects displayed and assists the user in finding and focusing on a specific object. The context of this pattern is broad and is not limited to visualization applications. It is applicable in our case and may help in fulfilling the user's need for a simple interface.

Furthermore, patterns # 9-14 which include *Multi Level Help*, *Wizard* and *Details on Demand* also address Serge's need for a simple interface, but more importantly, his need for reassurance by providing him with easily-accessible user support. First, Multi Level Help consists of providing light and heavyweight support to users with varying needs. In our case, this pattern helps the user to better understand the system, its capabilities, and the required actions to perform (general functionality, behavior and state). A user with a strong need for validation will benefit from this pattern [Tidwell 2005]. Secondly, the Wizard simplifies a task, following a structure and giving the reassurance that "things will turn out ok". Thirdly, Details on Demand reduces the complexity and amount of information presented to the user; it suggests that the user should be presented with only a summary of the information and allowed to obtain the details as part of a different display or as a tool tip.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Finally, let us give an example of a pattern that did not make it on the list. *Command History* is used when users need to keep a visible record of the actions they have performed. This pattern is applicable for more advanced users, and those who perform long and complex sequences of actions. The score of this pattern with Serge's user variables is "0", which fits with our understanding of this persona and the context of use.

6.3.3. Modifying selection parameters [Step 3b in Fig. 6-2]

The sub-steps from the previous section describe the most common path for pattern selection. However, an alternative path may occur after sub-step 10, where the designer is not satisfied with the recommended patterns. Therefore, the set of influential user variables initially selected may be modified based on the personas, after which the sub-steps are re-iterated.

Let us continue with our example. We decide that although the most highly recommended patterns from the list aim to reduce interface complexity, with secondary patterns aiming to provide the user with some reassurance, we need more patterns that provide support in the form of guidance for the user. Based on Serge's description, we add two more influential user variables to our current set: Guidance and education level. Serge's need for guidance (somewhat high, with a value of 3) came out strongly in our persona description, and his level of education (elementary, value of 1) further supports his need for a simpler interface with additional support in the form of guidance.

We re-iterate through the sub-steps, and the resulting patterns are illustrated in Table 6-13. Compared to the last selection, the set of recommended patterns continues to focus on the needs of users with low literacy (top three patterns), reducing complexity of the interface, and providing reassurance. These were all discussed in the last section. However, there are a number of differences which stand out. First, *Multi-level Help* (pattern #4) is now an even more highly recommended pattern compared to our previous selection. Secondly, *Navigation Page* (#11) and *Illustrated Choices* (#13) both suggest the use of pictures and metaphors in addition to written text, to either guide the user with navigation or with decision making. Our previous list only had

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the Illustrated Choices pattern. Thirdly but most importantly, we notice a set of patterns which primarily deal with user guidance: Patterns #14-17. For example, *Same-Page Error Messages* indicate messages by marking them on top of a page and making them extremely visible. Another example is *On-Fly Description*, which provides a descriptive sentence in close spatial proximity to the target object.

#	Title	Short description
1	Simple Text	Organize content using simple and short text
2	Alternative Orthography	Write content phonetically (phonetic spelling)
3	Audio Communication	Provide content in audio format
4	Multi Level Help	Mixture of help techniques to support varying needs
5	Filter	Filtering facilities to reduce number of data items
6	Stack Page	Use several surfaces stacked together to group content
7	Global Navigation	Consistent set of links to navigate to key sections
8	Sequential	Organize pages in a sequence (linear structure)
9	Hierarchical	Organize pages using a hierarchical cascade model
10	Details on Demand	Uses a separate window to display item details
11	Navigation Page	Use meaningful pictures/metaphors to hint user to proper area
12	Wizard	Step user through tasks and maintain visibility
13	Illustrated Choices	Use pictures instead of words to show choices
14	Same-Page Error messages	Mark top of page with message; indicate originating control
15	On-fly Description	Provide descriptive phrase in close proximity to target object
16	Input Hints	Place an explanation/example beside text field
17	Smart Selection	Automatically select coherent group of items
18	Contrasting Font weights	Separate levels of info by using contrasting font weights
19	Tiled Page	Divide the page into panes so that user can find relevant info
20	Overview plus Details	Place overview of information next to zoomed detailed view

Table 6-13: A modified subset of patterns with scores

To further demonstrate the type of information which the pattern list can provide, let us look at the *Details on Demand* pattern (#10). Recall that it reduces the amount of information by presenting only a summary of the information and allowing the user to obtain details as part of a different display or as a tool tip. It is essentially opposite in its philosophy to the *Overview plus Details* pattern, which is the last one (#20) on our pattern list. Overview plus Details suggests that the user should be presented with a small overview of the information. The user can then adjust and focus on a subsection of the larger detailed view on the same screen. The position of Details on Demand in the ordered pattern list is higher than Overview plus Details, which is close to the bottom of the list. In essence, this demonstrates that our rule-based scoring technique

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

recommends that users should be presented only with a simple and high level summary, allowing them to investigate further and obtain more details only when needed.

6.3.4. Filtering pattern set [Step 4 in Fig. 6-2]

This step entails filtering the pattern set according to the persona and design context information. A subset of relevant patterns is chosen for the pattern composition phase to follow; based on their applicability to the design context (including the domain and envisioned tasks of the application). Patterns which are obviously not suited for the design are omitted, thereby making the set of patterns to be more manageable.

One example of a pattern that might be suitable for filtering from Table 6-12 is *Reference Context* (#10). Its context of use (spatially locating and comparing objects) and the proposed solution (to provide a grid, plane etc.) is not applicable to our design. Although this pattern has a relatively high score, its context does not fit our envisioned website for low literacy users; there is no requirement for the user to accurately identify and compare individual objects while interacting with the application. A second example is *Input Hints* (#16) from Table 6-13. This pattern can also be omitted – since our website's goal is to provide basic information for low-literacy users, there will be no situations where users will be required to input information into a text field.

Finally, let us examine patterns with lower scores (for example, #16-20 and lower). Based on their position in the list, we might assume that these patterns are not important or that they should be omitted at this stage. In some cases, this would be premature. We should always keep in mind that the scoring system is only a guide in suggesting more relevant patterns based on the examined user group. Patterns with lower scores may still be applicable for a particular design, and it is up to the designer to consider if they address the needs of the examined user group.

6.4. Pattern Composition

This phase consists of one step: Composing patterns. This is a purely design activity where the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

selected patterns from the previous phase are used to create a design. Recall that a major advantage of patterns and their associated languages are their generative nature, meaning that they can essentially be combined together as building blocks. However, saying that, design is an activity dependent on each designer's creativity, background and expertise. Our goal is to simply provide some structure to the design activity, by presenting designers with: (1) a Pattern-Oriented Design (POD) model, and (2) a means to exploit pattern relationships. Artifacts such as task and interaction models may be used during this step, although they are external to our UX-P process. The designer iterates through various compositions until a satisfactory pattern-oriented design is attained.

6.4.1. Composing patterns [Step 5 in Fig. 6-2]

First, designers should follow a POD model. We have published literature on a preliminary version of this model [Javahery and Seffah 2002]. As part of this thesis, we refined the model and the corresponding pattern relationships. POD defines the overall design composition of a particular type of application, including a breakdown of this composition into different UI facets. The model acts as a guide for designers in making stepwise design decisions. To illustrate, for website design, we define four steps that designers should follow: (1) Defining the architecture of the site with architectural patterns, (2) Establishing the overall structure of each page with page manager patterns, (3) Identifying content-related elements for each page with information container patterns, and (4) Organizing the interaction with navigation support patterns. Landay and Myers [2001] and Welie [2003] also propose to organize their Web pattern languages according to both the design process and UI structuring elements (such as navigation, page layout and basic dialog style).

Secondly, designers should exploit relationships between patterns. We have described five types of relationships between the UPADE patterns, published in [Taleb et al. 2006; Javahery et al. 2006]. The same relationships can easily be applied to other pattern libraries. This multi-criterion classification is based on the original set of relationships [Zimmer 1995; van Duyne 2002; Yacoub and Ammar 2003] used to classify the patterns proposed in Gamma et al. [Gamma 1995].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The relationships are used to compose a UI design, allowing designers to make suppositions such as: "For some problem P, if we apply Pattern X, then Patterns Y and Z apply as sub-ordinates, but pattern S cannot apply since it is a competitor." The relationships are:

1. **Similar** (X, Y) if X and Y address the same problem within a similar context, by providing different solutions. As a result, X and Y can be replaced by each other in a certain composition. For example, index browsing and menu bar patterns are similar (see Figure 6-4). They both provide navigational support in the context of a medium size website, allowing users to navigate amongst items from the menu. Therefore, the index browsing pattern can be replaced by the menu bar pattern and still solve the same design problem. Moreover, because both patterns provide different solutions to the same problem, they can be used at the same time in a design. In our example below, each pattern is used for a distinct set of navigation items.

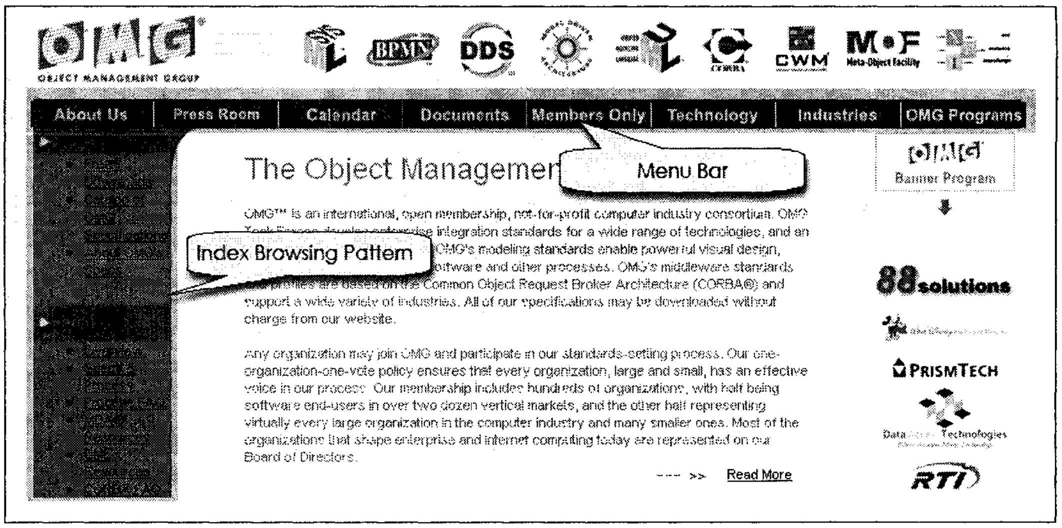


Figure 6-4: Comparison of Similar Patterns

2. **Competitor** (X, Y) if X and Y address the same problem within a similar context, by providing equivalent solutions. In other words, X and Y are competitors if they are similar and interchangeable. As a result, they cannot be used at the same time in a design. For example, the Web convenient toolbar and menu bar patterns are competitors (see Figure 6-5).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The convenient toolbar solution states: "Group the most common convenient action links, such as home, site map help etc." The convenient toolbar allows a user to directly access a set of common services from any Web page. At the same time, the menu bar pattern, when used as a shortcut, provides an equivalent solution: "Provide a collection of most frequently visited page links." Both patterns provide the same solution of presenting a group or a collection of most frequently used links. Hence, making these patterns competitors.

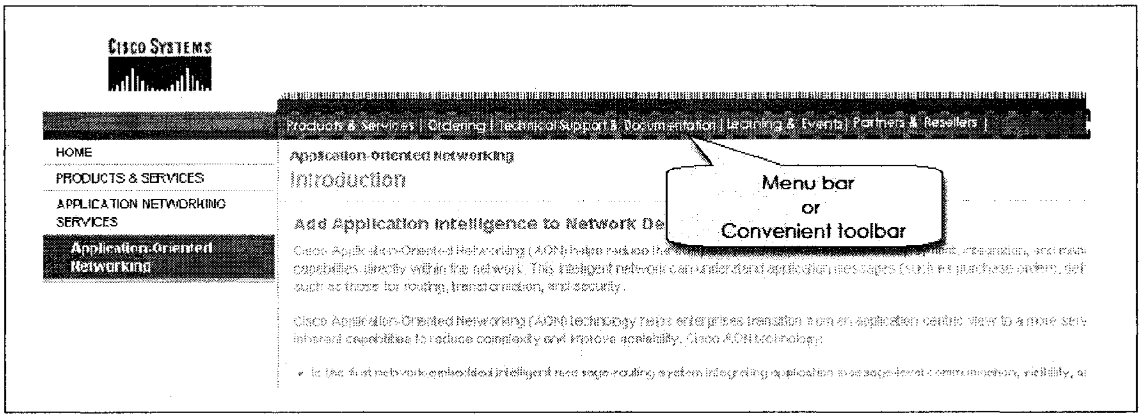


Figure 6-5: Two Competitor Patterns

3. **Super-ordinate** (X, Y) is a basic relationship to compose several patterns of different categories. A pattern X that is a super-ordinate of pattern Y means that Y is used as a building block to create X. For example, the home page pattern is a super-ordinate of convenient toolbar and index browsing patterns; because, both of them are used in home page pattern (see Figure 6-6).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

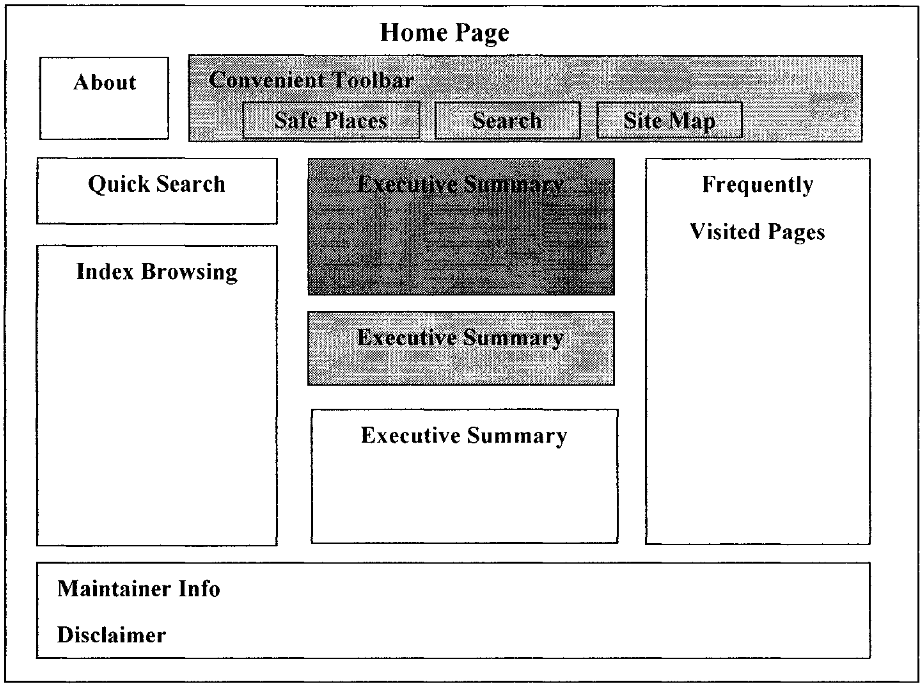


Figure 6-6: Home Page pattern with sub-ordinate patterns

4. **Sub-ordinate** (X, Y) if and only if X is embeddable in Y. Y is also called super-ordinate of X. This relationship is important in the mapping process of POD. For example, a Home Page pattern is composed of several other patterns, such as index browsing and convenient toolbar patterns (see Figure 6-6). All patterns used in a home page pattern will be sub-ordinate to it.

5. **Neighboring** (X, Y) if X and Y belong to the same pattern category (family). For example, the Sequential and Hierarchical patterns (Figure 6-7 and Figure 6-8) are neighboring because they belong to the category of Architectural patterns.

Within the scope of the development of web-based applications using the UPADE language (Appendix B), POD allows for the exploitation of 48 pattern relationships, allowing even novice developers to use the underlying best practices to iterate through concrete and effective design solutions. As described in our pattern model (chapter 5), each pattern contains a list of related patterns. For example, the Stack page pattern would contain the following information about related patterns: (1) Super-ordinate: Sequential, Hierarchical, Grid, Composite. (2) Sub-ordinate:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Executive Summary, On Fly Description, Browsing Index. (3) Competitor: Focus Page, Tiled Page. Let us illustrate how pattern composition can be applied to our website design for Serge. We will use our POD model in combination with the patterns selected from Table 6-13.

Step 1: Defining the architecture of the site with architectural patterns

The first step is to define the architecture of the site by applying architectural patterns. From Table 6-13, we see that two architectural patterns are recommended for Serge: Sequential and Hierarchical. The simplest architectural pattern is the Sequential pattern which organizes web application content as a sequence, or a linear narrative. Information that naturally flows as a narrative, time line or in logical order is ideal for this pattern. An example is an online tutorial. The Hierarchical pattern is a tree-based hierarchy, and is one of the best ways to organize complex bodies of web information. Hierarchical organization schemes are particularly well suited to organizing a complete web site. The user can easily go through from the most general overview of the web site, such as home page, down to the most specific or optional topics. These two patterns are illustrated in Figure 6-7 and Figure 6-8.

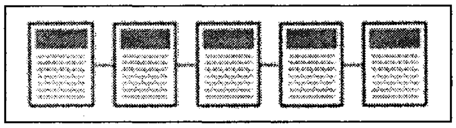


Figure 6-7: Sequential Pattern

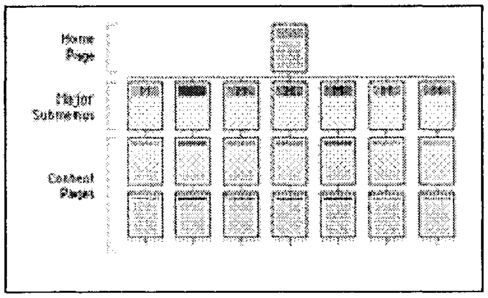


Figure 6-8: Hierarchical Pattern

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We cannot structure all our information as a linear narrative, and therefore the Sequential pattern does not fit our design context. We decide that a simplistic implementation of the Hierarchical pattern will better suit our website. Note that in both the pattern descriptions, the following relationships are indicated: (a) Grid pattern as neighboring, (b) Composite pattern as super-ordinate, and (c) Navigation page, Tiled page, Stack page amongst others, as sub-ordinate. If we had decided that neither the Sequential or Hierarchical patterns were useful for our design, we could have exploited pattern relationships to find an appropriate architectural pattern. For example, the Grid pattern is a neighboring pattern, which means it belongs to the same category of patterns. It should be used when topics and contents are fairly correlated with each other, and there is no particular hierarchy of importance. Procedural manuals, lists of university courses or medical case descriptions are often best organized using Grid patterns. For larger and more complex websites, a combination of all three patterns is often required, referred to as the Composite pattern. Note that this information is also contained within the relationships, where it is indicated as being super-ordinate to the Sequence, Grid and Hierarchical patterns.

To conclude this step, we choose the Hierarchical pattern for our example with Serge. After the pattern is chosen, it should be instantiated to the design. Specific pages should be given names and organized within the overall architecture.

Step 2: Establishing the structure of each page with page manager patterns

During the second step, the designer applies page manager patterns to establish a consistent physical and logical screen layout for each page that was defined in the previous step. The decision of which patterns to apply can follow in part from the relationship between architectural patterns and page manager patterns, where the former is a super-ordinate of the latter. From Table 6-13, we see that a number of page manager patterns are recommended for Serge: Stack Page, Navigation Page, and Tiled Page.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

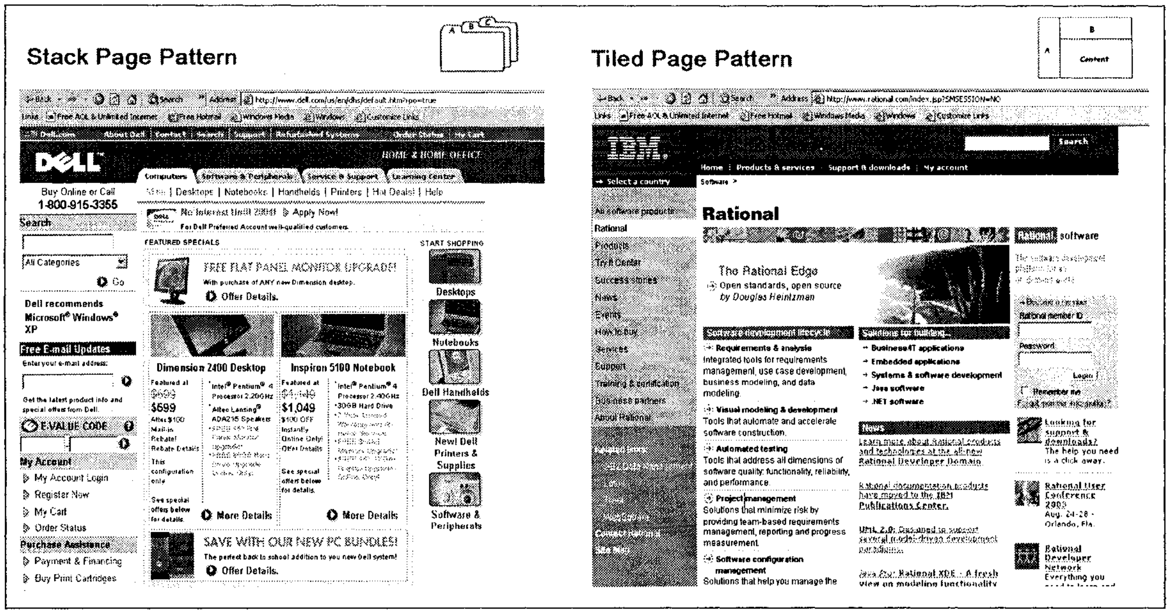


Figure 6-9: Comparison of Stack and Tiled Page Patterns

Furthermore, we consider relationships between competitor, similar and neighboring patterns. Again, from the selected patterns for Serge, we note some interesting relationships. First, the Stack and Tiled Page patterns (see Figure 6-9) are competitors. This means that if you choose the Tiled Page to design your home page, you cannot use the Stack Page for any of the subsequent pages. Such knowledge can be critical for pattern users because if it is not taken into consideration during design, it can compromise the benefits of the pattern. Secondly, the Navigation Page pattern is only neighboring and not a competitor to either the Stack or Tiled Page patterns, and can therefore be used simultaneously in the design with either one of them. We note that in our selected list of patterns, the Stack Page pattern has a higher score relative to the Tiled Page pattern. This is logical when considering Serge's needs for a simple interface.

To conclude this step, we only choose the Navigation Page pattern for our example with Serge. This user group will benefit from the use of pictures and metaphors instead of more complex structural page organizations. The Navigation Page pattern is illustrated in Figure 6-10.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

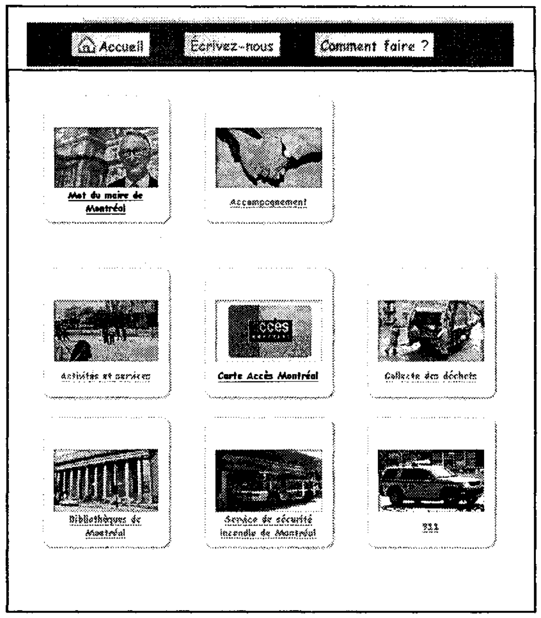


Figure 6-10: Use of Navigation Page Pattern in [Montreal City 2006]

Step 3: Defining content structure with information container patterns

The third step involves employing information container patterns, to plug in an information segment for each page. Patterns from this category contain best design practices about how to structure and define content for website pages. Similarly to the previous step, the decision of which patterns to apply can follow in part from the relationship between page manager patterns and information container patterns, where the latter is a sub-ordinate of the former. For example, sub-ordinate patterns indicated for the Navigation Page pattern include the Executive Summary pattern. The Executive Summary pattern summarizes contents by using concise sentences beneath or beside underlying topics. Before the Web was invented, authors of technical documents recognized that readers appreciate short segments of information. Such design practices should be embedded in the design process and presented to the designer, which is the precise aim of POD. The Executive Summary pattern provides a solution by providing an information preview. Users can then decide if they wish to view the entire document depending on whether the information is relevant. Figure 6-11, explained in the next section, illustrates an example of how the Executive Summary pattern can be used within a design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

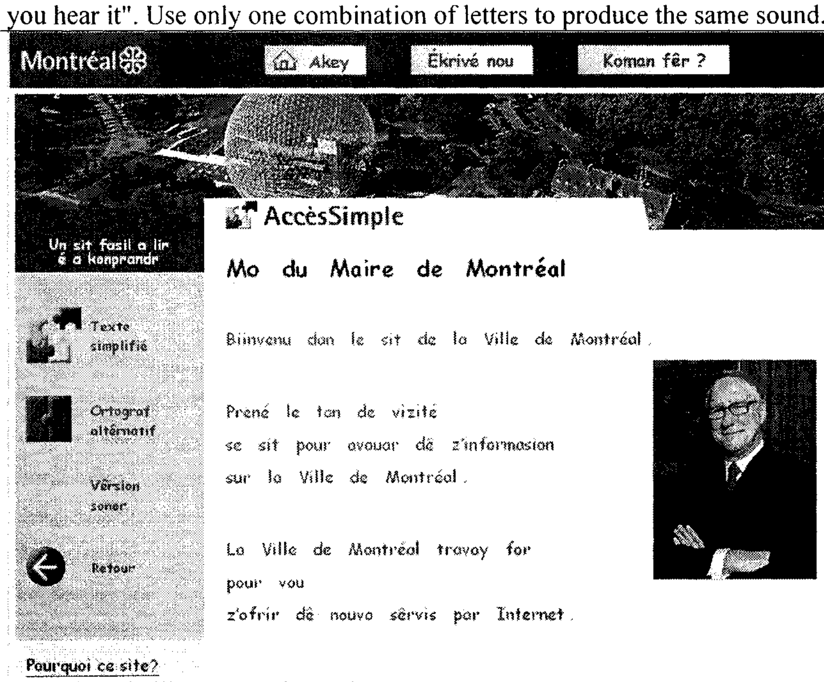
In the case of Serge, we decide that the Executive Summary pattern is not necessary since we will rely on pictures and metaphors to convey choices, with a concise title underneath each illustration. However, from Table 6-13, we realize that our three special needs patterns – Simplified Text, Alternative Orthography, and Audio Communication – fit well with the content needs of our website. To illustrate, the Alternative Orthography pattern offers simplification of text by using phonetic spelling. Alternative writing uses only 35 relations between letters whereas the conventional French language uses more then 4000. It should be noted that alternative writing is not a new way to write in French. It is a different way of presenting the language, similar to how brail is used for the blind. This pattern is described in Table 6-14. Note that we have discovered and documented all the low literacy patterns ourselves, based on the City of Montreal website [Montreal City 2006]. We have used the site as a guide to construct three patterns which can be reused for individuals with a low literacy level, and/or with cognitive disabilities. The rest of the patterns are illustrated in Appendix E. The work is based on guidelines referenced in [Montreal City 2006].

Step 4: Applying navigation support patterns to facilitate navigation in information spaces

The fourth and final step consists of building the navigation support. Since there is no hierarchical relationship between Navigation Support patterns and the previously mentioned categories, it is possible to consider navigation elements earlier in conjunction with other patterns. Navigation support patterns suggest different models for navigating between information segments and pages. We examine two patterns: Index Browsing and Global Navigation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

<table>
<tr><th>Title</th><th>Alternative Orthography</th></tr>
<tr><td>Context</td><td>The user has serious difficulties in reading and understanding information because of limited intellectual capacity or low literacy.</td></tr>
<tr><td>Problem</td><td>How to reduce the complexity of the medium used (language) to communicate information without necessarily impeding the content.</td></tr>
<tr><td>Forces</td><td>may require simplified written communication; may require spelling phonetically</td></tr>
<tr><td>Solution</td><td>Reduce the complexity of the written language by simplifying the orthography. Construct an alternative writing style with a simple orthography with a goal to allow for "write it as you hear it". Use only one combination of letters to produce the same sound.</td></tr>
<tr><td>Examples</td><td>



from [http://ville.montreal.qc.ca]

- Simplifies French orthography by reducing the amount of relations between letters from 4000 to 35.

</td></tr>
<tr><td>Related Patterns</td><td>Neighboring: Simplified Text, Audio Communication.</td></tr>
</table>

Table 6-14: Alternative Orthography Pattern

First, let us examine Index Browsing, which is used to navigate amongst a list of content pages. We decide to look at this pattern because it is one of the navigation patterns closely related to other patterns we considered for our website design; it is sub-ordinate to the Hierarchical pattern, and super-ordinate to the Executive Summary pattern. Furthermore, Executive Summary, combined with Index browsing, allows users to preview information about a certain topic before spending time to download, browse and read different pages (Figure 6-11). Executive Summary is weighted as a highly recommended sub-ordinate pattern when pattern users try to use the Index

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

browsing pattern. Knowledge about context-oriented relationships, as described above, can be very useful to pattern users. They can be a guide in choosing the best solution for a specific user problem based on a particular context.

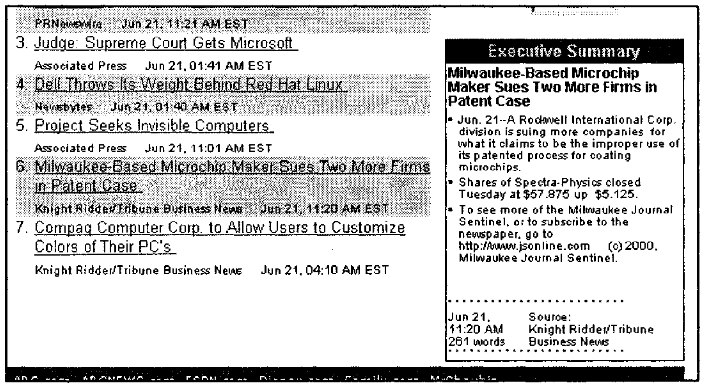


Figure 6-11: Index Browsing and Executive Summary Patterns

Secondly, we examine Global Navigation. Global Navigation came up in our selected list of patterns for Serge, and is used to provide a consistent set of links for facilitated navigation to key sections of the website. Global Navigation and Index Browsing are neighboring patterns since they belong to the same category of patterns. Although they are both used for navigation support, they are not used to solve the same usability problem and are applied in different contexts. Global Navigation gives users an overview of the higher-order structure of the UI consistently on each page, facilitates exploration and easy movement to key sections. Index Browsing provides a list of links to content pages, often on the left-hand side of the home page, and allows the user to reach these pages promptly. For Serge, we decide that our website will benefit from the facility in movement that Global Navigation provides. This is illustrated in Figure 6-12.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

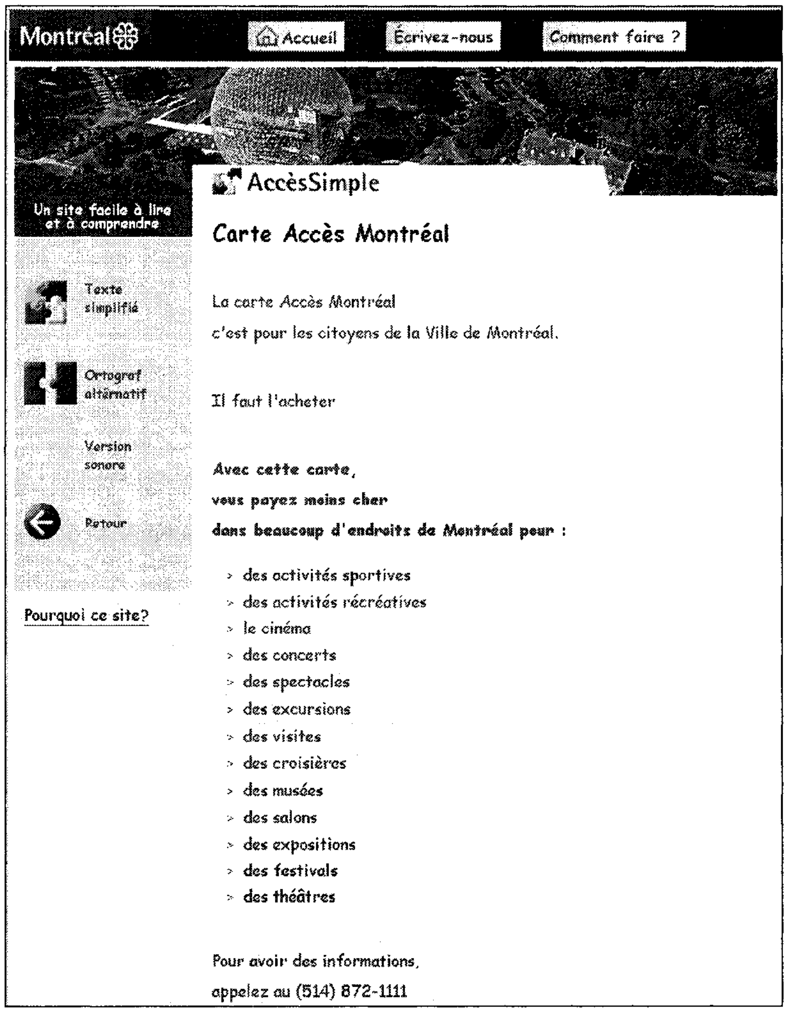


Figure 6-12: Use of Global Navigation pattern in [Montreal City 2006]

Similarly to the above, the step of composing patterns can be iterated as many times as needed; until the designer is satisfied with the resulting design. Some steps may need to be iterated more than others. For example, the number of page manager patterns will be greater than the number of architectural patterns for most designs, and therefore, more iterations are typically needed. By composing patterns like "building blocks", a comprehensive design begins to take shape.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6.5. Summary of the UX-P Process

In this chapter we detailed our proposed UX-P Process. The process consists of three phases and a set of associated steps that support designers in the derivation of a pattern-oriented design from personas. The essence of the process is to first cluster users based on most influential user variables. During this phase, user data provided by the designer is clustered into a set of personas. Clustering is performed on the basis of selected user variables, which are deemed significant by the designer.

In addition to hidden dependencies discovered during persona creation, these variables drive a rule-based scoring technique that provides a suggested set of patterns for the designer's consideration. The confidence of each rule is determined empirically, and their summation is used to compute scores, resulting in a final recommendation. The proposed scoring system bridges the gap between user variables and the actual patterns by defining two intermediate constructs: User needs and pattern criteria. The need for these intermediate criteria was elicited through expert input. The designer then filters the candidate patterns according to the design context.

In the last phase of our process, the designer composes patterns into a comprehensive design. Pattern composition is guided by both a POD model and various interrelationships that exist among patterns. Moreover, it is guided by other significant design artifacts such as task models, use cases and/or given design standards.

In the next chapter we present the P2P Mapper, a supporting environment consisting of a set of tools to assist the designer in carrying out some of the activities involved in the UX-P process.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 7

Operationalization and Validation

In this chapter, we detail a walkthrough of our process within the context of a real-life design project. First, we briefly describe the P2P Mapper Tool which we implemented to support designers using our process. Secondly, we describe a case study with the UI development of a Bioinformatics application.

7.1. Tool Support: P2P Mapper

As an attempt to better assist designers in using our process, we developed a supporting tool for putting the process into practice (operationalization). As illustrated in the previous chapter, the proposed process involves a set of repetitive, tedious and time consuming tasks. In addition, some of the steps and artifacts described in the process have been constructed in a format which allows for automation. We therefore built a tool called the Persona to Pattern (P2P) Mapper.

The general steps comprising our process are illustrated in Table 7-1. The persona creation and pattern selection phases were amenable for partial automation. In particular, we automated the following steps: Clustering users (step 1) and selecting patterns based on personas (step 3). Moreover, we provided features for users to carry out the remainder of the persona creation and pattern selection phases.

No.	Steps	Tool support	Input	Output
Persona Creation				
1	Clustering users	Y (automated)	user data	user clusters
1b	Modifying clustering parameters	Y	user clusters	modified parameters
2	Refinement of persona set	Y	user clusters	personas
Pattern Selection				
3	Selecting patterns based on personas	Y (automated)	personas	pattern set
3b	Modifying selection parameters	Y	pattern set	modified parameters
4	Filtering pattern set	Y	pattern set	filtered pattern set
Pattern Composition				
5	Composing patterns	N	filtered pattern set	pattern-oriented design

Table 7-1: UX-P Process Steps and Tool Support

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The P2P Mapper provides the designer with three major features: (1) the data entry system, (2) the clustering utility, and (3) the pattern selection utility. The data entry system provides the user with an interface to enter, view and modify user information. In particular, the designer provides a set of discrete user variables; optionally he/she may also include narrative text illustrating popular user scenarios and other textual descriptions. All user information is stored in XML format as described in chapter 5.

Once the data is entered, the tool provides the designer with automatic and interactive clustering capabilities to derive quantified persona specifications. Clustering is performed based on the discrete user variables provided during the data entry phase. The tool provides the user with the choice between two clustering techniques, namely K-Means clustering and Interactive clustering. The former technique is performed fully automatically but requires, a-priori, an indication of the desired number of clusters. The later technique is performed in an interactive manner, where the designer selects a subset of the user variables, on which the clustering should be based on. The tool then returns a number of clusters, which can be iteratively refined and reduced by further constraining the allowed range of values of the selected user variables.

Automatic clustering is more suitable for novice designers as it can be used as a black box technique where the required user intervention is minimal. It leaves however the designer with very little control to influence the outcome of the clusters. Iterative clustering is an interactive clustering method, which mimics the designer's strategy of manually building personas. Hence it provides the user with more influence capability but requires an advanced knowledge of the user variables and their domain on which the clustering will be based on. At this point we note that if none of the automatically or interactive clustering methods result the in the desired set of clusters, the tool provides the designer with the option to manually manipulate and enter the various clusters.

Once a quantified persona set is created, the designer can use the mapping module in order to create a set of patterns. This step is performed automatically based on the set of personas

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

generated during the previous step. In particular the mapping module selects patterns using the scoring system described previously, where based on a set of rules, the various patterns are associated with scores. After the successive application of all the rules, a set of patterns ordered by highest score is generated. The designer can then filter and reorganize the set based on additional design criteria, such as: user scenarios, domain, pattern type, source library, etc.

Figure 7-1 depicts the architecture of the P2P Mapper. In its current version, the tool supports the first two phases of the UX-P process, namely persona creation and pattern selection. We also note that our tool comes with a pre-populated library containing the formalization of 83 patterns from the following pattern libraries: GUI [Tidwell 2002], web [Javahery and Seffah 2002; Welie 2003] visualization [Wilkens 2003], and our own "special needs" patterns.

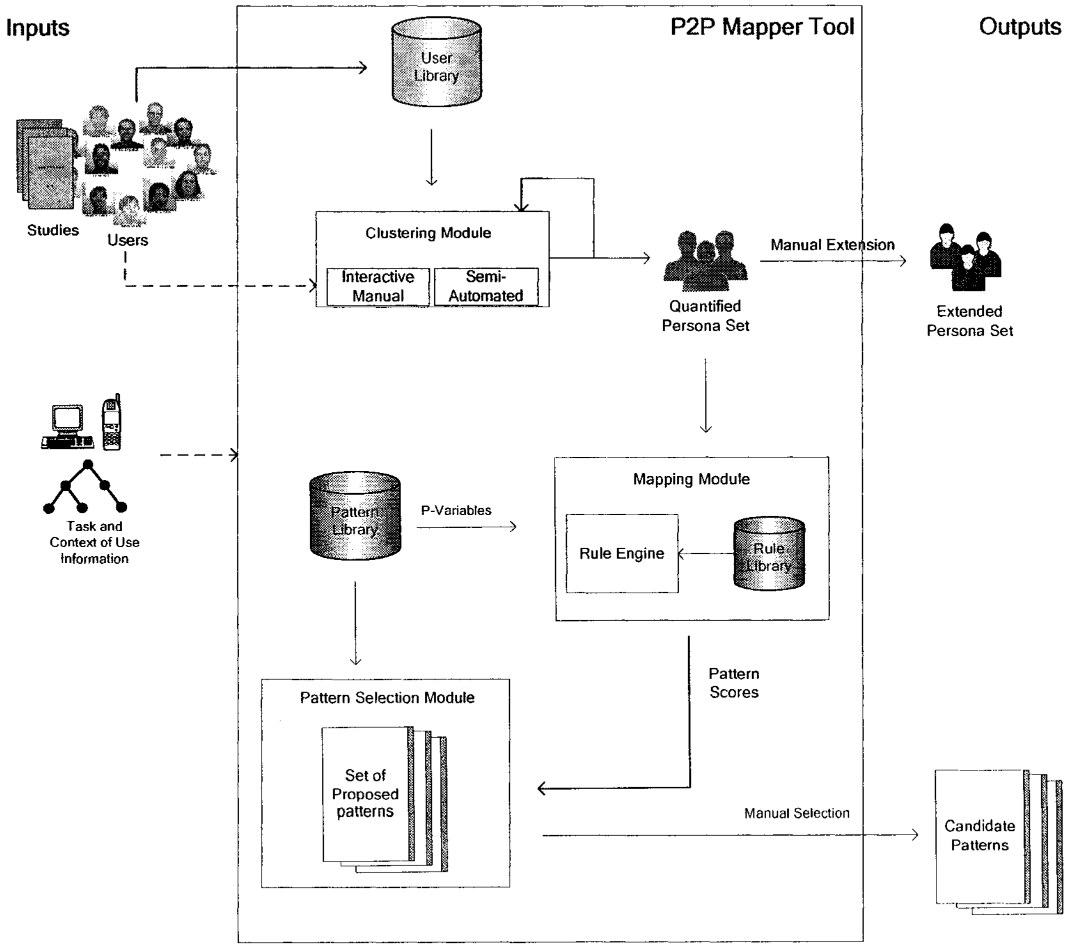


Figure 7-1: Overview of P2P Mapper Tool

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For further information about the tool and the algorithms used, the reader is referred to the Master's thesis of Alexander Deichman [2006].

7.2. Validation: Case Study 2

We conducted a validation study with a 3D Bioinformatics visualization tool, called Protein Explorer. The study first entailed the development of a prototype following the UX-P Process, and then testing the prototype with end-users. The goals of the study were to (1) assess the applicability of using the UX-P Process and P2P Mapper tool within the context of a HCI project and (2) to evaluate whether the UX-P process leads to more usable systems. A summary of the experimental technique and planning steps can be found in Appendix H.

We applied the UX-P Process, resulting in a conceptual design which we then used as a blueprint to build a test prototype. To test this new design, we carried out a comparative randomized study using task-based evaluation and open-ended interviews. It is important to note that as part of our study protocol, we did not carry out any empirical studies or pre-design analyses (including task model creation) with the original Protein Explorer tool, but with *similar* visualization tools[20]. Furthermore, empirical studies were conducted in the form of pre-design usability inquiries to gather information about users' interaction behaviors and needs.

7.2.1. Domain and Users

The selected domain and users are similar to the NCBI study. Therefore, the reader is referred to Section 3.2.1 for more details. The application we redesigned is called Protein Explorer [Protein 2005], a web-based software application for biomedical research, used for the prediction and analysis of bimolecular structure. Users explore various macromolecule structures such as Protein and DNA in 3D, using a web browser.

The Protein Explorer browser interface (see Figure 7-2) is split into four windows, organized as

---

[20] This was to ensure that the new design was only developed based on our process, without any possible positive "side-effects" due to the re-iteration and/or the discovery of usability issues with the original version of the tool.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

panes. The window on the right is the visualization interface, containing the 3D macromolecule. The structural data for this molecule comes from a Protein Data Bank and users can view molecules by entering their Protein Data Bank ID. The upper left window provides information about the molecule and includes display options. Furthermore, it splits into two windows with tips and hints in a "child" frame. There are links on this window that lead you to other resources. The lower left window of Protein Explorer is a message box detailing the atoms clicked in the visualization, and allows users to type commands in a scripting language called chime. According to [Protein Explorer 2005], the original tool was designed with a focus on "functionality and usability".

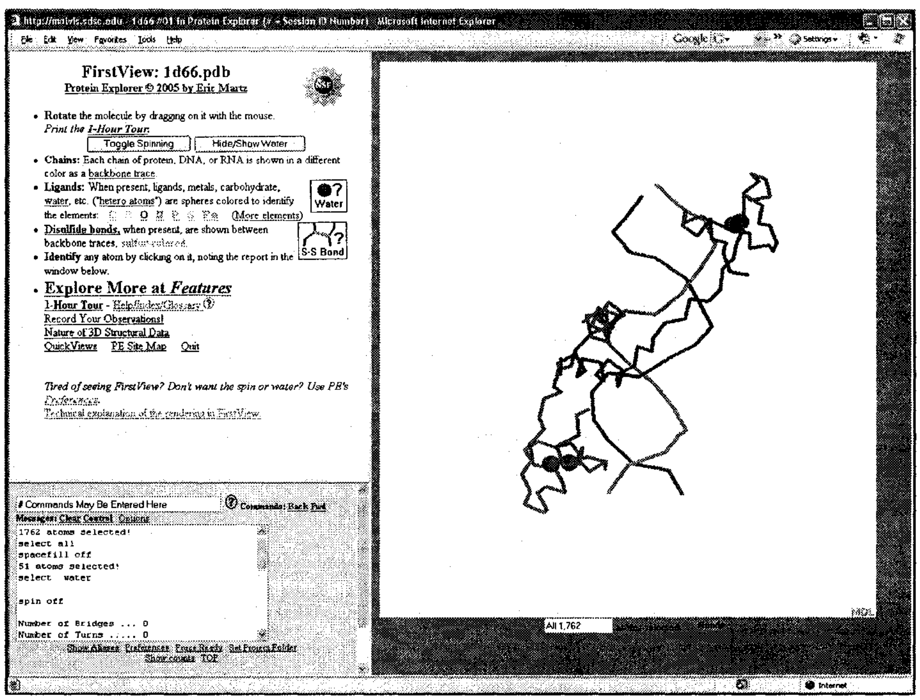


Figure 7-2: Protein Explorer Application

To better understand the user-tasks involved for such 3D Bioinformatics visualization tools, we constructed a task model. We also used this model as a guide during pattern selection and composition (see Figure 7-3).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

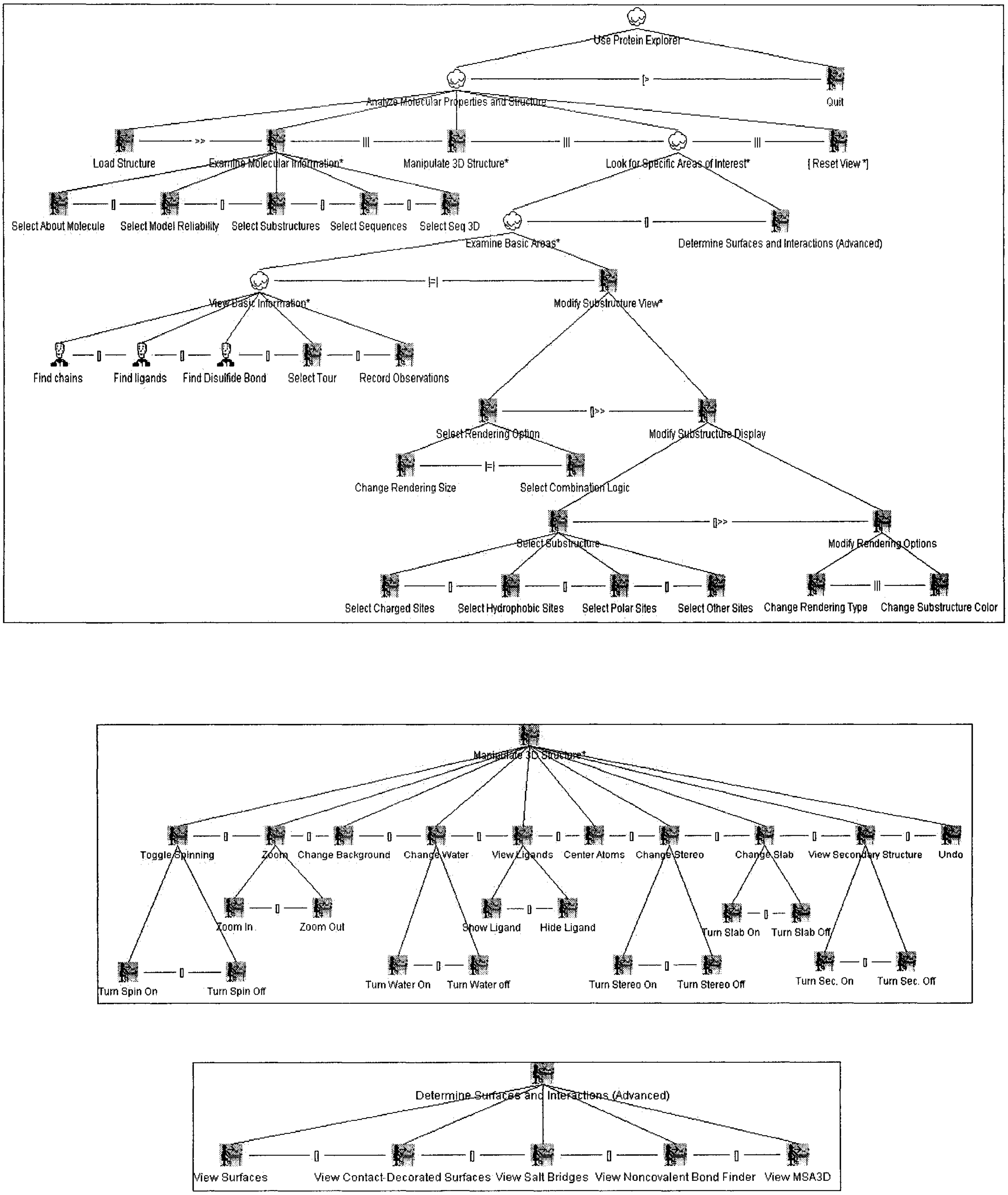


Figure 7-3: Task Model for 3D Visualization Tool

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

7.2.2. Applying the UX-P Process

We used a sample of 22 users from the biomedical-related fields to create personas. An aggregate description of their demographics, as user variables, is presented in Table 7-2.

User Variable	Value	
Age**	Mean	3.33
	SD	0.62
Gender	Male	8
	Female	14
Computer Experience*	Mean	2.91
	SD	1.15
Domain Experience*	Mean	2.09
	SD	1.38
Education Level **	Mean	5.55
	SD	0.51
Bioinformatics Experience*	Mean	2.00
	SD	1.20
Product Experience*	Mean	1.68
	SD	1.36

Table 7-2: Aggregate description of 22 participants of the study [21]

For each participant, a complete set of user variables (as described in the persona model) was recorded. Some user variables, such as education level, were recorded based on an initial questionnaire administered to participants. Others, such as learning speed, were recorded during user observations. Furthermore, we noted information about goals and interaction details for each user, and typical scenarios for a subset of the most representative users. This information was applied later to our personas. We carried out usability inquiries in the form of field studies and user observations on two Bioinformatics visualization tools, Cn3D [2005] and ADN-Viewer [Gros et. al. 2005]. We noted the following differences in interaction behaviors and dependencies between user variables:

- Users with medium and high domain experience were more feature-keen.
- Users with significant product experience had higher expectations in terms of both features and

[21] *the scales used are 0 to 4, where 0 means none and 4 means expert
**the scales used are 0 to 6; refer to persona model for precise meanings

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

performance, and were reluctant to learn a new design paradigm. This was especially apparent with individuals who came from computational backgrounds.

- Most users with high product experience either had high domain experience or were from computational backgrounds.
- Users with low and medium product experience often seemed confused when interacting with either tool, since the features were not sorted according the task model.
- The biologists needed more control when interacting with the tool. They were extremely dissatisfied when processes were automated. They wanted to understand how the automation worked. Biologists had a more experimental problem-solving strategy, where they followed a scientific process and were repeat users of specific features.
- Users from computational backgrounds had a more linear problem-solving strategy where they performed tasks sequentially. They also exhibited comfort when trying out new features.

As a result of the above information, we clustered our users based on the *domain experience* and *background* variables. We used the clustering tool from our P2P Mapper Environment (see Figure 7-4) to perform this step. Domain experience was a user variable from our persona model and therefore pre-defined in P2P Mapper. Note that it is related to experience in the Bioinformatics field (not to be confused with Biology). To make clustering manageable, we restricted the domain to three values: Low (0 and 1), Medium (2), and High (3 and 4). Background was an additional variable which we added as a parameter in the tool; its values were defined as being either “biology” or “computer science”.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

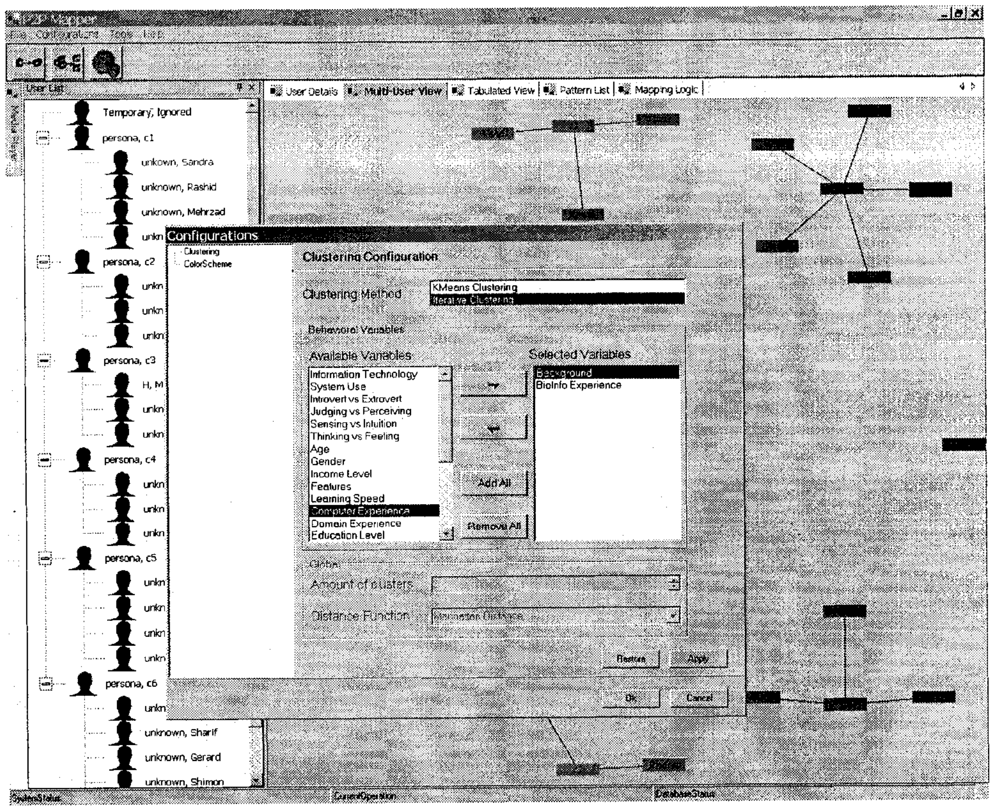


Figure 7-4: Clustering Tool in P2P Mapper Environment

The result of the clustering exercise was a set of six clusters. In order to refine this set, we carried out further user observations, where users were selected based on the clustering exercise. The result of this study showed that the interaction behavior of biologists with low domain experience were exactly the same as computer scientists. We also noticed that computer scientists with low domain knowledge did not need such an advanced tool. Moreover, we identified a new variable which we had not considered previously, *age:*

- Older users (45+) were more anxious when interacting with the system, and were less comfortable in manipulating the visualization.
- Older users had a high need for validation of decisions. They would often ask their assistants or others to help them in performing more complex tasks.
- Older users were often feature-shy.
- As age increased, the expectation of tool support increased. This is due in part to a decrease in learnability with older users.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

As a result, we eliminated two of the six clusters and repeated the clustering step after adding the variable *age*. Our user groups consisted of the following age groups: Young adults (20-34 years), mature adults (35-59 years), and seniors (60-74 years). We restricted the domain to young (20-45 years) and older (45+ years) users based on our observations. This second exercise resulted in eight clusters with the P2P Mapper tool. The results are illustrated in Table 7-3.

#	Clustering results (domain experience, background, age)	Decision
C1	Med domain, computer, old	remove
C2	Med domain, computer, young	keep
C3	High domain, computer, old	remove
C4	High domain, computer, young	remove
C5	High domain, biology, old	keep
C6	High domain, biology, young	remove
C7	Low domain, biology, old	keep
C8	Low domain, biology, young	remove

Table 7-3: Results after Modification of Clustering Parameters

We eliminated five out of the eight clusters since their attributes and behaviors were contained within other clusters. First, C1, C3 and C4 were removed because their functional needs were satisfied by C2. When considering individuals with a computer science background, *age* and the variation between *medium/high domain experience* did not seem to notably influence interaction behavior. Secondly, C6 was removed because it was satisfied by C5 and C2. Although the potential learnability of C6 is higher than C5, their functional needs were satisfied by C5. Therefore, if we satisfy C5 than we can satisfy C6. In terms of being feature-keen and comfortable with system interaction, C6 acts like C2. Thirdly, we removed C8 because it was satisfied by C7 and C2. The expectations in terms of functional needs of C7 are similar to C8. However, since the learnability of C8 is higher, if we satisfy C7 then we can satisfy C8. Furthermore, in terms of being more comfortable with system interaction (such as manipulation of the visualization), C8 acts like C2.

Based on C2, C5 and C7, we constructed the following set of personas by using the original user descriptions as a guide: (1) Marta Aviles, a young bioinformatics professional working in

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

industry, (2) Zhang Hui, a senior Parasitology professor, and (3) Sue Blachford, a mature adult who is a medical practitioner with limited experience in Bioinformatics. To exemplify, Marta Aviles is illustrated in Figure 7-5. A detailed description of the persona can be found in Appendix G.

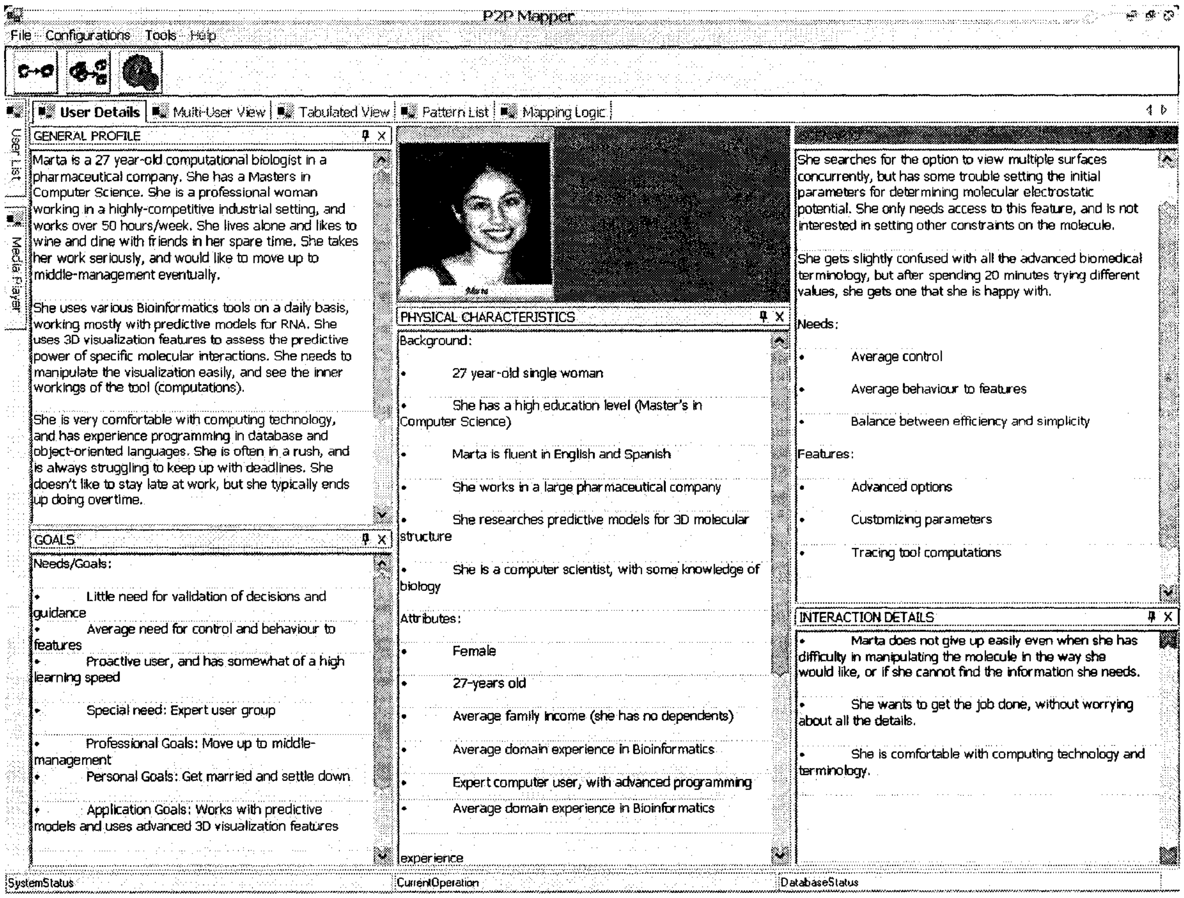


Figure 7-5: P2P Mapper view of the Persona Marta

Based on our constructed personas, we chose a set of user variables for pattern selection. The identifiers, hidden dependencies, and values used for pattern selection are illustrated in Table 7-4.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Personas	Identifiers	Hidden dependencies and specific needs	Values for pattern selection
#1: Marta Aviles 	- med domain - computer background - young adult	- Average control and behavior to features - Balance between efficiency & simplicity	Special need = expert Age = 3 Behavior to features = 2 Control = 2 Domain experience = 2
#2: Zhang Hui 	- high domain - Biology background - older adult	- Feature-keen and needs lots of control - Needs validation of decisions	Special need = colorblind Age = 5 Behavior to features = 4 Control = 4 Domain experience = 4
#3: Sue Blachford 	- low domain - biology background - middle-age adult	- Feature-shy and needs guidance - Needs simplicity	Special need = novice Age = 4 Behavior to features = 0 Control = 0 Domain experience = 1

Table 7-4: User Variables for Pattern Selection

Each persona resulted in a different set of patterns. In particular, we noticed that the selected patterns between persona 2 and 3 varied the most, whereas persona 1 contained patterns from both personas. We decided that a compromise solution would best fit our purpose. We therefore redesigned the tool using the following patterns: Button Groups (1), Card Stack (2), Good Defaults (3), Legend (4), Multi-level help (5), Details on Demand (6), Tool Tips (7), Convenient Toolbar (8), Action Panel (9), Command History (10), Filter (11), and Reduction Filter (12). The numbers in brackets correspond to the numbers as indicated in Figure 7-6 and Figure 7-7. Note that following the same reasoning as during our empirical studies and pre-design analyses, we did not aim to "fix" specific usability problems with the original tool. This was so we could more precisely compare the two designs based on the application of our design process. The only requirement we had was to keep the same "look and feel" for the design, including the use of several panes.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

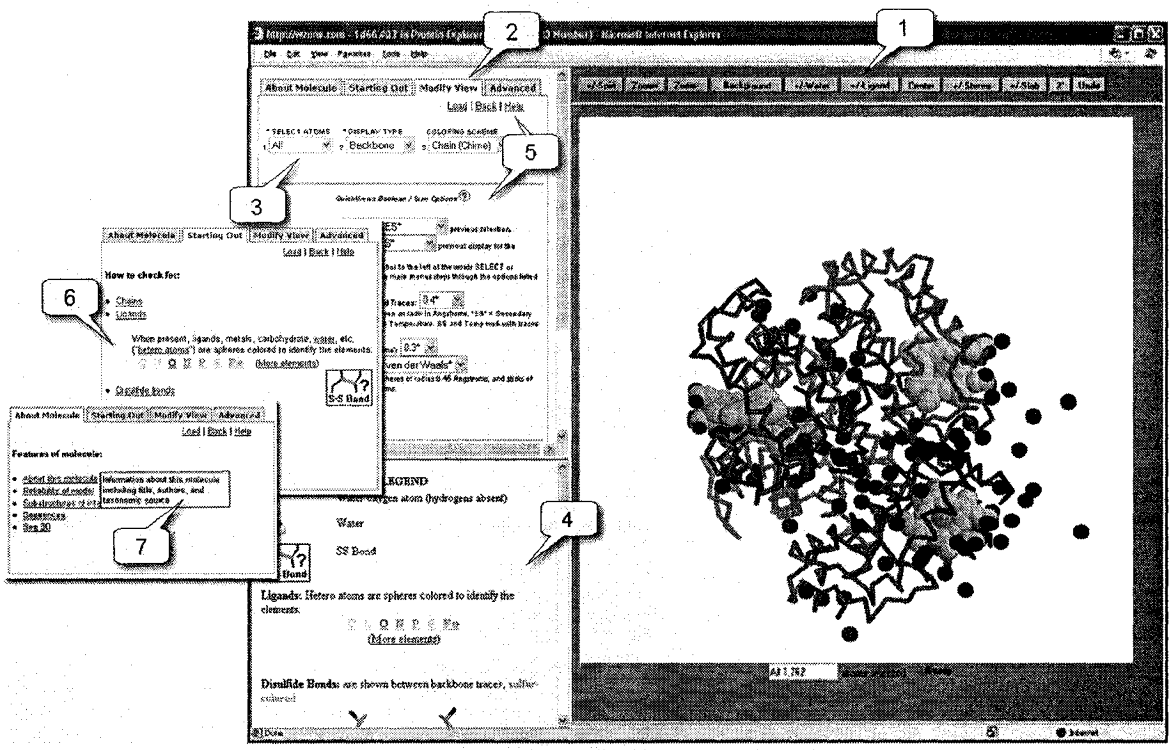


Figure 7-6: Protein Explorer prototype designed using UX-P Process (first set of patterns)

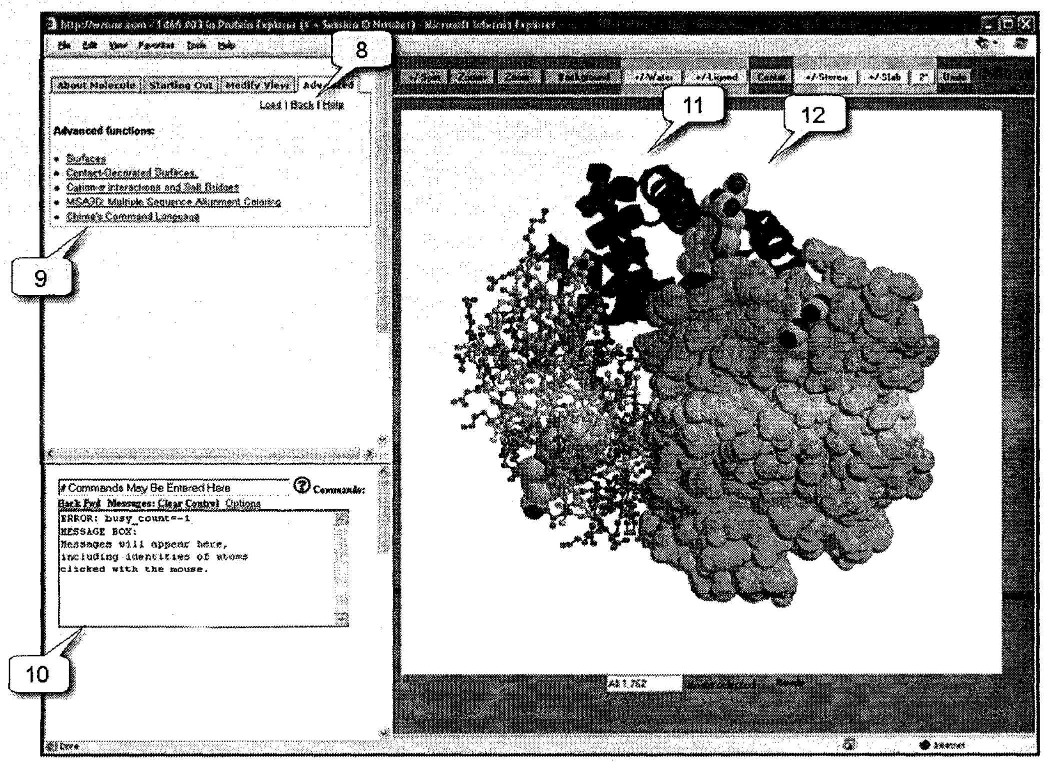


Figure 7-7: Protein Explorer prototype designed using UX-P Process (second set of patterns)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

7.2.3. Testing the Prototype with End-Users

We used a sample of 15 end-users from the biomedical-related fields for usability testing. Our sample was a subset of the users that participated in our pre-design activities. It is also important to note that although some of our users had experience with bioinformatics visualization tools, none of them had any experience with the Protein Explorer. This was advantageous for us, since there was no transfer of learning effects from expert users. Furthermore, they were unaware of which version of the tool (original vs. new) they were interacting with during the sessions. An aggregate description of the user demographics, as user variables, is presented in see Table 7-5.

User variables	Values	
Age**	Mean	3.27
	SD	0.55
Gender	M	4
	F	11
Computer Experience*	Mean	3.00
	SD	1.07
Domain Experience*	Mean	2.13
	SD	1.30
Education Level **	Mean	5.6
	SD	0.51
Bioinformatics Experience*	Mean	1.73
	SD	1.03
Product Experience*	Mean	1.73
	SD	1.22

Table 7-5: Aggregate description of 15 participants of the study [22]

Method

We performed task-based evaluations and open-ended interviews to compare the original design with the new design. Open-ended interviews included general questions about impressions of both versions of the tool (any differences, likes and dislikes) and specific questions about the user interface (navigation, etc.). Tasks were designed in conjunction with a biomedical expert. End-users of the tool typically follow a scientific process when performing tasks; i.e., the exploration

[22] *the scales used are 0 to 4, where 0 means none and 4 means expert
** the scales used are 0 to 6; refer to persona model for precise meanings

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of a particular molecule. We therefore designed each task as part of a scientific process. One example is presented below. The rest of the tasks can be found in Appendix B.

A. Exploring the Hemoglobin molecule

1. Load Hemoglobin structure, with the PDB code 1HGA.
2. Stop the molecule from spinning.
3. Remove the ligands from the molecule.
4. Modify view to "spacefill" (from "backbone") for all atoms of the molecule.
5. Find out more about this molecule. For example, title and taxonomic source.
6. In advanced options, find out how to view surfaces (multiple surfaces concurrently). This will allow you to use options such as "molecular electrostatic potential".

We used a *within-subjects* protocol, where each user performs under each condition; in our case, each user tested both designs, the Original Design (Design O) and the New Design (Design N). The advantage of this protocol is that there is less of a chance of variation effects between users, and we can obtain a large data set even with a smaller number of participants. In order to reduce the effect of learning, we varied the order of the designs [Dix et al. 2003] per participant – some users started with Design N, others with Design O. Furthermore, we varied each of the two scientific processes per design type. We logged task times, failure rates, and the recorded the entire user experience with both designs.

Results

First, we present the **quantitative results** (see Table 7-6 and Table 7-7). For the purposes of the statistical analysis, we have two independent variables: (1) Variation of the design type and (2) variation of the design order used. Dependant variables are: (1) Task time and (2) failure rate. However, we expect that the second independent variable has no effect on the results. More precisely, we expect that by effectively varying the starting type of the design we have been able to reduce any effect of knowledge transfer between the designs to a minimum.

For **task duration**, we used the ANOVA[23] test in order to compare task times of Designs O and

[23] An explanation of this test was detailed in Chapter 3, and will not be repeated here.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

N. Our hypothesis for the test was that we would have a statistically significant improvement of time required to complete a task in Design N compared to Design O. We performed an ANOVA two- factor test with replication in order to prove our hypothesis. The two factors selected where: (1) the order in which the user tested the designs (ON or NO) and (2) the design type tested (O or N). The goal of this test was to see whether each factor separately has an influence on the results, and at the same time to see if both factors combined have an influence on the design.

Source of Variation	**F**	**P-value**	**F crit**	η^2
Factor 1	2.024175	0.167682	4.259675	0.03
Factor 2	35.70645	3.62E-06	4.259675	0.55
Interaction	3.182445	0.087084	4.259675	0.05

Table 7-6: Results for task duration

The results demonstrate that variation of the order in which the user has tested the design has no influence on the task times ($p > 0.05$). This means that the users were unaffected by transfer of knowledge from one design to another. Moreover, the test demonstrates that the combined effect of both variables has no statistically significant impact on the task times ($0.05 < p < 0.10$). Finally, the second factor is the only one that has a statistically significant effect on the task times: $F = 35.71$, $p = 3.62$ E-06, $\eta^2 = 0.55$. This demonstrates that there was a statistically significant improvement in task time in Design N when compared to Design O. We noted an average improvement of **52 %.**

For **failure rates**, our hypothesis was that there should be a significant improvement in failure rates with Design N versus O. Similar to task times, we performed a two factor ANOVA test with replication in order to test our hypothesis, where the factors were the same as described above.

The test results demonstrated that there was a statistically significant improvement in failure rates in Design N when compared to Design O: Factor 2 has $F = 28.03$, $p < 0.05$ and $\eta^2 = 0.49$. Moreover, the test demonstrated that there is no statistically significant interaction between the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

two factors when considering their effect on failure rates ($p > 0.05$). Similarly, the test has demonstrated that the order under which the users have tested the designs has no statistically significant effect on the failure rates ($p > 0.05$).

Source of Variation	**F**	**P-value**	**F crit**	η^2
Factor 1	4.033333	0.055991	4.259675	0.07
Factor 2	28.03333	1.97E-05	4.259675	0.49
Interaction	0.833333	0.37039	4.259675	0.01

Table 7-7: Results for failure rate

The **qualitative results** were obtained from open-ended interviews with all users, carried out after task-based evaluations with both versions of the tool. The most common comments about the usability of the original version from end-users were as follows: (1) it is overloaded with content in the control pane; (2) the provided information is not filtered adequately, requiring users to spend lots of time reading irrelevant information, (3) navigation between pages is difficult, resulting in confusion when trying to reach the load page; and (4) manipulation of the visualization pane is difficult since it is unclear where the features for the visualization are located. Furthermore, we recorded the sessions and used the think-aloud protocol with users. Our observations indicated a high level of frustration with users during their interaction with the original version of the tool.

The most common comments about the usability of the new prototype from end-users were as follows: (1) easier to locate information because of the structure; (2) organization of features and tools follows more closely with the scientific process in bioinformatics, (3) the interface is simpler and users feel more in control when interacting with it, (4) the use of tabs made navigation easier. Furthermore, during the recorded sessions, users seemed calmer and more comfortable during their interaction with the prototype.

13 out of 15 users indicated that they prefer the design of the new prototype compared to the design of the original tool. Simplicity and "feeling more in control" were cited as the most

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

important reasons. Interestingly enough, one of the two users who indicated his preference for the original tool also cited "simplicity" as a reason, but in terms of the new prototype as being **too** simple, and the original version having all the information "handy". The other user indicated that the fonts were too small and the colors a bit confusing on the new prototype.

7.3. Summary of Case Study 2

In this chapter we presented the P2P Mapper Tool, which we implemented to support designers using our process. The tool partially automates the persona creation and pattern selection phases. We used our tool in a second case study that we carried out with the UI development of a Bioinformatics visualization application. The goals of the study were to assess the applicability of using the UX-P Process and P2P Mapper tool within the context of a HCI project, and to evaluate whether the UX-P process leads to more usable systems.

We achieved the first goal by applying our process step-by-step in the derivation of a new conceptual design of a Bioinformatics tool called Protein Explorer. According to the first phase of the UX-P process, we first used clustering to create three personas, which were derived from a set of clusters automatically suggested by the P2P mapper. In the second phase, we used the P2P Mapper to generate a list of candidate patterns. From the list, we selected applicable patterns and composed them, resulting in a new conceptual design (phase 3).

To address the second goal, we carried out a comparative study with our prototype and the original tool. Quantitative results indicated both a statistically significant improvement in task duration and failure rates with the prototype. Furthermore, qualitative results indicated a greater degree of satisfaction with the prototype for 13 out of the 15 users.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 8

Conclusion

In this thesis we propose a UI design method, with an associated process and tools, to support designers in deriving a conceptual design from user experiences. In current practice, the derivation of a conceptual design from user experiences is based on loosely-defined guidelines, giving rise to a significant "gap" between user requirements and design outcomes. This is especially problematic for novice designers, who cannot rely heavily on their design experience.

Our UI design method consists of three phases, namely Persona Creation, Pattern Selection and Pattern Composition. Furthermore, the method is based on a set of key UCD principles which we have enriched with "engineering-like" concepts such as reuse and traceability. Details of the method are in part extracted from our experiences during Case Study 1; where a design prototype was built using personas and patterns as the primary design directives. The application of interest was a Bioinformatics website, which biologists use as a portal to access different analytical tools. The new design was compared to the original design, and resulted in significant improvements in terms of usability.

Following this method, we propose a systematic process called the UX-P Process and an associated tool. The process rigorously defines a set of steps from persona creation to the composition of a comprehensive design. Based on knowledge elicited from HCI experts, it incorporates a clustering step as part of persona creation, and a set of rules to select patterns from persona specifications. Furthermore, we propose more formal representations for personas and patterns amenable for tool support. As part of Case Study 2, we tested our process and tools in the design of a Bioinformatics visualization tool. Our prototype was compared to the original tool, resulting in significant improvements in terms of usability measures.

To this end, the contributions of this thesis are as follows:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1. *The definition, development and validation of a practical method (process and tools) to support HCI designers and UI developers in the derivation of a conceptual design based on user experiences. The method:*
   - Addresses a crucial problem in user-centered design by narrowing the well-known gap between user requirements and user interface design; more specifically between user experiences and conceptual design. This is achieved by defining a methodical link through the use of user variables, pattern variables, and usability principles.
   - Defines a systematic process (called the *UX-P Process*) that involves moving from user experiences represented as personas to a pattern-oriented design, in a series of steps. The conceptual design is composed of patterns, which are selected based on persona specifications.
   - The process is traceable since any given conceptual design is composed of patterns, and for any given pattern, a set of user needs can be identified.

2. *Identification of the key phases and steps of the UX-P Process.* The phases consist of Persona Creation, Pattern Selection, and Pattern Composition. Each phase is associated with a set of steps and decision points that have been explicitly defined and illustrated with key examples. Designers may iterate within each phase before proceeding onto the next phase. In addition, each phase can be used in isolation or within a different process, since they provide solutions for common problems involved in UI design.

3. *Formalization, systematic representation, and modeling of user experiences.* This includes the (1) Characterization of a subset of user experiences as variables with discrete values. These variables determine variations in interaction behavior and needs, and are amenable for analysis and tool support, (2) Definition of a clustering technique and associated tool to help designers cluster a large set of users based on significant user variables, interaction behaviors and needs. This results in user groupings which can then be transformed and represented as personas, (3) Capturing users and their experiences in a persona model which includes both formalizations of user experiences in discrete terms and informal descriptions of scenarios and user goals. The latter provides background and context of use information to designers.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The combination of both formal and informal descriptions will guide designers during the design process, in decision making, and tradeoffs to be made. The formal descriptions are amenable for automatic analysis by a software tool, acting as input for the pattern selection rules.

4. *Classification, extension and formalization of patterns.* Pattern descriptions are represented by a pattern model which characterizes patterns by: (1) Classifying them, based on the usability design principles they address, within categories called pattern criteria. (2) Extending pattern descriptions to include knowledge about user-centered and usability design principles. This includes information about the target user group, domain, and pattern criteria. (3) Formalizing the most relevant parts of pattern descriptions into a pattern header file, amenable for further automatic analysis by a software tool.

5. *Pattern selection based on a set of rules.* Pattern selection is based on a set of rules, inferred from dependencies between user variables, user needs, pattern criteria, and patterns. These were identified through knowledge elicitation techniques carried out with HCI experts. The rules were implemented within a scoring system which takes user variables as input and outputs a list of patterns, ordered by their relevance.

6. *Prototypical tool support for designers.* We have implemented a prototypical tool to provide support for our design process. The *P2P Mapper* provides the designer with tool support for the persona creation and pattern selection phases, and automates two of the sub-steps. An interactive environment is provided for the designer where she/he can enter user data, as well as view and modify both personas and candidate patterns.

7. *Empirical validation supporting the contributions including the process and the tools.* In HCI and empirical software engineering, user-oriented studies are required both to motivate the research as well to assess the validation and accuracy of the proposals. Two large empirical studies were carried out with end-users to test and validate the process. The first study was a proof-of-concept, intended for testing purposes and process discovery. The second study was

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

used to validate the use of the process and the associated rules. Both studies were comparative in nature, and tested qualitative and quantitative factors. Recall our hypothesis, which was that our design process would result in more usable systems, satisfying the collection of user experiences. Both our empirical studies indicated a positive increase in usability measures for our design prototypes, including a significant improvement in task times and user satisfaction.

Our research resulted in an initial design process to derive conceptual designs from user experiences; however, it led to additional research questions that can be avenues for further research.

First, we described two types of associations between users and patterns, through their needs: A direct association with special needs, and an indirect association with usability needs. These associations allow designers to select a set of patterns appropriate to their design. An interesting possibility for investigation would be to further filter patterns based on task type. Recall that pattern descriptions make reference to typical tasks of the user-task model. Some patterns are only applicable for a particular task type (i.e. *Advanced Search Pattern* and tasks of type *search*). These task types could act as further input into pattern selection, and could even be included in pattern headers.

Secondly, both our persona and pattern models are a good starting point in standardizing the representation of personas and patterns, respectively. In current practice, personas are constructed based on general narrative guidelines and contain information which allows them to be little more than a communicative tool. Furthermore, pattern writers have few guidelines in constructing patterns. This results in little consistency in the structuring and definition of pattern descriptions for pattern libraries. It would be interesting to explore the use of both models as a standard representation of these two artifacts in HCI.

Thirdly, the associations discovered between user variables, needs, and pattern criteria were based on knowledge elicited from HCI experts. It would be valuable to test each one of these

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

through experimentation with end-users, to determine their precise impact. In this vein, we note that our rule-based scoring system is based on an approximation of the heuristics elicited during expert consultation. Our knowledge elicitation activities concentrated on establishing trends and rates of change. A further improvement of our scoring system can be its enhancement with learning capabilities. Hence, the current implementation of the rules can be understood as an initial configuration of a future expert system or a neural network, which may be further adjusted and fine tuned based on the assessment of the quality of the results.

Finally, we tested our approach within the context of two case studies. Further experimentation with other domains of application and with designers is necessary to refine and improve our approach. The case studies presented in this research work were carried out with web-based and visualization applications. Depending on the existence of pattern libraries, our process is flexible enough to be applicable to other domains. Moreover, further testing of our process with designers will be an important next step. This will allow for practical research and industry feedback, which is crucial in determining the refinements required to improve and effectuate the adoption of such a pattern-oriented design framework in practice.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

References

Aaker D.A., V. Kumar and G.S. Day. 2004. *Marketing Research, 8th edition*. New York: John Wiley and Sons.

Abran, A., Khelifi, A., Suryn, W. and Seffah, A. 2003. Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*. **11** (4).

Alexander, C. 1979. *The Timeless Way of Building*. New York: Oxford University Press.

Alexander, C., S. Ishikawa, S. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel. 1977. *A pattern language - Towns, buildings, construction*. New York: Oxford University Press.

Apple Inc. 2003. *Apple Human Interface Guidelines*. [Accessed August 2005]. Available from: <http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/>

Bastien, J.M.C. and D.L. Scapin. (Unpublished, 1993). *Ergonomic Criteria for the Evaluation of Human-Computer Interaction*. Technical Report N. 156. INRIA, France.

Baxley, B. 2003. Universal model of a user interface. *In: Proceedings of the 2003 Conference on Designing for User Experiences (DUX '03)*. June 6-7, 2003, San Francisco, California. New York: ACM Press. pp. 1-14.

Bevan, N. 1995. Measuring usability as quality of use. *Software Quality Journal*. **4**, pp.115-140.

Bevan, N. 1997. Quality and usability: A new framework. *In*: V.E. Veenendaal and J. McMullan (eds). *Achieving software product quality*. Netherlands: Tutein Nolthenius.

Bevan, N. and M. Macleod. 1994. Usability measurement in context. *Behavior and Information Technology* **13**, pp.132-145.

Borchers, J.O. 2001. *A Pattern Approach to Interaction Design*. New York: John Wiley and Sons.

Byrne, M.D., and A. Kirlik. (Unpublished, 2003). *Integrated Modeling of Cognition and the Information Environment: A Closed-Loop, ACT-R Approach to Modeling Approach and Landing With and Without Synthetic Vision System (SVS) Technology*. Technical Report AHFD-03-4/NASA-03-3. Aviation Human Factors Division, University of Illinois, Urbana-Champaign, Savoy, IL.

Carroll, J.M. 2000. *Making Use: Scenario-Based Design of Human-Computer Interactions*. Cambridge, MA: The MIT Press.

Cn3D. 2005. *Cn3D 4.1 Download and Homepage* [online]. [Accessed October 2006]. Available from: http://130.14.29.110/Structure/CN3D/cn3d.shtml

Cognetics Corporation. 2003. *Persona of a Cognetics Design Specialist* [online]. [Accessed June, 2005]. Available from: <http://www.cognetics.com/about/team/people4.html >

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Constantine, L.L. and L.A.D. Lockwood. 1999. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. New York: ACM Press.

Cooper, A. 1999. *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. Indianapolis: SAMS Publishing.

Cooper, A. and R.M. Reimann. 2003. *About Face 2.0: The Essentials of Interaction Design*. Indianapolis: John Wiley and Sons.

Coram T. and J. Lee. 1998. *Experiences: A Pattern Language for User Interface Design* [online]. [Accessed January 2002]. Available from: <http://www.maplefish.com/todd/papers/experiences>

Courage, C. and K. Baxter. 2005. *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*. San Francisco: Elsevier.

Deichman, A. 2006. *The Persona to Pattern Mapper*. Master's Thesis in Computer Science (In Progress), Concordia University, Montreal.

Dix, A., J.E. Finlay, G.D. Abowd and R. Beale. 2003. *Human-Computer Interaction, 3rd Edition*. Upper Saddle River, NJ: Prentice Hall.

Dufresne, A., and S. Turcotte. 1997. Cognitive style and its implications for navigation strategies. *In*: B. Boulay and R. Mizoguchi (eds.), *Artificial intelligence in education knowledge and media learning system*. Amsterdam: IOS Press.

Ehn, P. 1988. *Work-Oriented Design of Computer Artifacts*. Stockholm: Arbetslivscentrum.

Engelberg, D. and A. Seffah. 2002. Design Patterns for the Navigation of Large Information Architectures. *In: Proceedings of 11th Annual Usability Professional Association Conference*. July 8-12, 2002, Orlando, Florida.

Erickson T. 2000. Lingua Francas for Design: Sacred Places and Pattern Languages. *In: Proceedings of Designing Interactive Systems*. August 17-19, 2000. New York: ACM Press.

Fenton, N. E. and S.L. Pfleeger. 1998. *Software Metrics: a Rigorous and Practical Approach, 2nd Edition*. Boston, MA: PWS Publishing Co.

Gamma, E., R. Helm, R. Johnson and J. Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Gould, J.D., S.J. Boies, and C. Lewis. 1991. Making usable, useful, productivity-enhancing computer applications. *Communications of the ACM*. **34** (1), pp.74-85.

Gros, P.E., N. Férey, J. Hérisson, R. Gherbi and A. Seffah. 2005. Multiple User Interface for Exploring Genomics Databases. *In: Proceedings of HCI International*. July 22-27, 2005, Las Vegas, USA.

Henninger, S., Haynes, K. and M. Reith. 1995. A framework for developing experience-based usability guidelines. *In: Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, & Techniques*. August 23-25, 1995, Ann Arbor, Michigan. pp.43-53.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Holtzblatt, K. and S. Jones. 1995. Conducting and Analyzing a Contextual Interview. *In*: R.M. Baecker, J. Grudin, W.A.S. Buxton and S. Greenberg (eds.). *Readings in Human-Computer Interaction: Toward the Year 2000*. San Francisco, CA: Morgan Kaufman.

ISO Standards. 1991. ISO 9241: 1991. *Standard on Guidance on Usability.*

ISO Standards. 1998. ISO 13407: 1998. *Standard on Human-Centered Development Processes for Interactive Systems.*

ISO Standards. 2000. ISO TR 18529: 2000. *Standard on Human-Centered System Development.*

ISO Standards. 2006. ISO/IEC 25062: 2006. *Software product Quality Requirements and Evaluation (SQuaRE), Common Industry Format (CIF) for usability test reports.*

Ivory, M. Y. and M. A. Hearst. 2001. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.* **33** (4), pp.470-516.

Jacobson, I. 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach.* New York: ACM Press.

JAVA. 2006. *Java Look and Feel Design Guidelines* [online]. [Accessed June 2006]. Available from: <http://java.sun.com/products/jlf/>.

Javahery, H., and A. Seffah. 2002. A Model for Usability Pattern-Oriented Design. *In: Proceedings of TAMODIA 2002.* July 18-19 2002, Bucharest, Romania, pp. 104-110.

Javahery, H., A. Seffah, D. Engelberg, and D. Sinnig. 2003. Chapter 12: Migrating User Interfaces across Platforms Using HCI Patterns. *In*: A. Seffah and H. Javahery (eds.). *Multiple User Interfaces*. Chichester, UK: John Wiley and Sons.

Javahery, H. 2003. *Pattern-Oriented Design for Interactive Systems*, Master's Thesis in Computer Science, Concordia University, Montreal.

Javahery, H., A. Seffah, and T. Radhakrishnan. 2004. Beyond power: making bioinformatics tools user-centered. *Communications of the ACM* **47** (11).

Javahery, H., D. Sinnig, A. Seffah, P. Forbrig and T. Radhakrishnan. 2006. Pattern-Based UI Design: Adding Rigor with User and Context Variables. *In: Proceedings of TaMoDia 2006.* October 23-24, 2006, Hasselt, Belgium.

Kim, C. and J. Kim, J. 2003. A Recommendation Algorithm Using Multi-Level Association Rules. *In: Proceedings of the IEEE/WIC international Conference on Web intelligence.* October 13-17, 2003, Halifax, Canada.

Kirakowski, J., and M. Corbett, M. 1993. SUMI: The Software Measurement Inventory. *British Journal of Educational Technology.* **24** (5), pp. 210-212.

Konstan, J. 2001. Heavyweight Applications of Lightweight User Models: A Look at Collaborative Filtering, Recommender Systems, and Real-Time Personalization. *In: User Modeling 2001: 8th International Conference Proceedings, UM 2001.* July 13-17, 2001, Sonthofen, Germany.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Laakso, S. 2003. *User Interface Design Patterns* [online]. [Accessed November 2003]. Available from: <http://www.cs.helsinki.fi/u/salaakso/patterns/>

Lafrenière, D. and A. Granlund. *A Pattern-Supported Approach to User Interface Design, UPA 99 Workshop Report* [online]. [Accessed December 2002]. Available from: <http://www.gespro.com/lafrenid/Workshop_Report.pdf>

Landauer, T, K. 1997. Behavioral Research Methods in Human-Computer Interaction. *In:* M.G. Helander, T. K. Landauer, P. Prabhu, and P. V. Prabhu (eds.). *Handbook of Human-Computer Interaction, 2nd Edition*. New York: Elsevier Science Inc. pp.203–227.

Landay, J.A., and B.A. Myers. 2001. Sketching Interfaces: Toward More Human Interface Design. *IEEE Computer*. **34** (3).

Mao, J., K. Vredenburg, P.W. Smith, and T. Carey. 2001. User-centered design methods in practice: a survey of the state of the art. *In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research*. November 5-7, 2001, Toronto, Canada.

Mayhew, D. 1999. *Usability Engineering Lifecycle*. San Francisco, CA: Morgan Kauffman.

McGraw K and K. Harbison-Briggs. 1989. *Knowledge Acquisition Principles and Guidelines*. New Jersey: Prentice Hall.

McGrenere, J. 2002. Iterative design and evaluation of multiple interfaces for a complex commercial word processor. *In*: A. Seffah and H. Javahery (eds.). *Multiple User Interfaces*. Chichester, UK: John Wiley and Sons.

McGrenere, J., R.M. Baecker and K.S. Booth. 2002. An evaluation of a multiple interface design solution for bloated software. *In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves*. April 20-25, 2002, Minneapolis, Minnesota. New York: ACM Press. pp. 164-170.

McKenzie, K. 2000. *10 Usability Heuristics* [online]. [Accessed March, 2003]. Available from: <http://www.studiowhiz.com/_publications/10heuristics.php>

Microsoft. 2004. *Official Guidelines for User Interface Developers and Designers* [online]. [Accessed Aug, 2006]. Available from:
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/welcome.asp>

Montreal City. 2006. [online]. [Accessed October 2006]. Available from:
<http://ville.montreal.qc.ca>

Morae. 2006. *Morae Usability Testing for Software and Web Sites* [online]. [Accessed Aug. 2006]. Available from: < http://www.techsmith.com/morae.asp>

Muller, M.J. and M. Czerwinski. 1999. Organizing usability work to fit the full product range. *Communications of the ACM*. **42** (5), pp.87-90.

Muller, M.J., D.M Wildman, and E.A White. 1993. Taxonomy of PD Practices: A Brief Practitioner's Guide. *Communications of the ACM*, **36** (6), pp.24-28.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Muylwijk, B.V., G. Veer, and Y. Waern. 1983. On the Implications of User Variability in Open Systems: An Overview of the Little We Know and of the Lot We Have to Find Out. In *Behavior and Information Technology*, **2** (4), pp.313-326.

Myers, B. 1995. User Interface Software Tools. *ACM Transactions on Computer-Human Interaction*. **2,** pp.64-103.

Myers, B. and M.B. Rosson. 1992. Survey on User Interface Programming. *In: Proceedings of Human Factors in Computing Systems (CHI 1992).* May 3-7, 1992, Monterey, CA. New York: ACM Press. pp. 195-202.

Najjar, L.J. 2000. *Conceptual User Interface: A New Tool for Designing E-Commerce User Interfaces* [online]. [Accessed April, 2005]. Available from: <http://www.internettg.org/newsletter/dec00/article_cui.html >

NCBI. 2005. *National Center for Biotechnology Information* [online]. [Accessed October 2005]. Available from: <http://www.ncbi.nlm.nih.gov/>

Nielsen, J. 1994. Heuristic Evaluation. *In*: J. Nielsen and R.L. Mack (eds.). *Usability inspection methods.* New York,: John Wiley and Sons.

Nielsen, J. 2001. *How to Conduct a Heuristic Evaluation* [online]. [Accessed November 2001]. Available from: <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

Norman, D.A. and S.W. Draper (eds.). 1986. *User Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Padda, H. 2003. *QUIM Map: A Repository for Usability/Quality in Use Measurement.* Master's Thesis in Computer Science, Concordia University, Montreal, Canada.

Potosnak, K. 1999. Conceptual design. *In: CHI '99 Extended Abstracts on Human Factors in Computing Systems.* May 15-20, 1999, Pittsburgh, Pennsylvania. New York: ACM Press. pp. 153-154.

Preece, J., Y. Rogers and H. Sharp. 2002. *Interaction Design: Beyond Human-Computer Interaction.* New York: John Wiley and Sons.

Protein Explorer. 2005. *FrontDoor to Protein Explore*r [online]. [Accessed October 2006]. Available from: < http://molvis.sdsc.edu/protexpl/frntdoor.htm>

Pruitt, J. and J. Grudin. 2003. Personas: practice and theory. *In: Proceedings of the 2003 Conference on Designing for User Experiences (DUX '03).* June 6-7 2003, San Francisco, California. New York: ACM Press. pp. 1-15.

Seffah, A., J. Gulliksen and M. C. Desmarais. 2005. *Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle.* The Netherlands: Springer.

Seffah, A. and H. Javahery. 2002. On the Usability of Usability Patterns, Patterns in Practice: A Workshop for UI Designers. *In: Conference on Human Factors in Computing Systems (CHI 2002).* April 20-25, 2002, Minneapolis, Minnesota.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Shneiderman, B. 1998. *Designing the User Interface, 3rd Edition*. Reading, MA: Addison Wesley. pp. 124-150.

Sinnig, D., A. Gaffar, D. Reichart, P. Forbrig and A. Seffah. 2004. Patterns in Model-Based Engineering. *In: Proceedings of Fourth International Conference on Computer-Aided Design of User Interfaces (CADUI 2004)*. January 14-16, 2004, Funchal, Portugal. pp. 197-210.

Sinnig, D., H. Javahery, P. Forbrig and A. Seffah. 2005. Patterns and Components for Enhancing Reusability and Systematic UI Development. *In: Proceedings of HCI International*. July 22-27, 2005, Las Vegas, Nevada.

Sommerville, I. 2000. *Software Engineering, 6th Edition*. Boston, MA: Addison-Wesley.

Statscanada. 2006. [online]. [Accessed September 2006]. Available from:
< http://www.statcan.ca/>

Sutcliffe, A. 2000. On the effective use and reuse of HCI knowledge. *ACM Transactions on Computer-Human Interaction*. **7** (2).

Tahir, M. F. 1997. Who's on the other side of your software: Creating user profiles through contextual inquiry. *In: Proceedings of Usability Professionals Association Conference (UPA '97)*. August 1997, Monterey, California.

Taleb, M., H. Javahery and A. Seffah. 2006. Pattern-Oriented Design Composition and Mapping for Cross-Platform Web Applications. *In: Proceedings of DSV-IS 2006*. July 26-28, 2006, Dublin, Ireland.

Tetzlaff, L. and D.R. Schwartz. 1991. The use of guidelines in interface design. *In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*. April 27 - May 2, 1991, New Orleans, Louisiana.

Thovtrup, H. and J. Nielsen. 1991. Assessing the usability of a user interface standard. *In: Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. April 27 - May 2, 1991, New Orleans, Louisiana.

Tidwell, J. 2002. *UI Patterns and Techniques* [online]. [Accessed March, 2003]. Available from: <http://time-tripper.com/uipatterns/index.php>

Tidwell, J. 2005. *Designing Interfaces*. O'Reilly Media, Inc.

UPADE. 2004. *Usability Pattern-Assisted Design Environment* [online]. [Accessed September 2006]. Available from <http://hci.cs.concordia.ca/www/hcse/projects/upade/>

van Duyne, D.K., J.A. Landay and J.I. Hong. 2003. *Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Boston, MA: Addison-Wesley.

Vanderdonckt, J., E. Furtado, J. Furtado and Q. Limbourg. 2003. Multi-Model and Multi-Level Development of User Interfaces. *In*: A. Seffah and H. Javahery (eds.). *Multiple User Interfaces*. Chichester, UK: John Wiley and Sons. pp. 193-216.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Verplank B., J. Fulton, A. Black and B. Moggridge. 1993. Tutorial on Observation and Invention: Use of Scenarios in Interaction Design. *In: Proceedings of INTERCHI'93*. April 24-29, 1993, Amsterdam. New York: ACM Press. pp.36.

Welie M.V. 2003. *Interaction Design Patterns* [online]. [Accessed March, 2003]. Available from <http://www.welie.com>

Welie M.V. and H. Trætteberg, H. 2000. Interaction Design Patterns in User Interfaces. *In: Proceedings of PLOP'2000*. August 13-16, 2000, Monticello, Illinois.

Welie M.V., G.C. Van der Veer and A. Eliëns. 1999. Breaking down Usability. *In: Proceedings of INTERACT 1999*. August 30 - September 3, 1999, Edinburgh, Scotland.

Welie, M.V., G.C. Van der Veer and A. Eliëns. 2000. Patterns as Tools for User Interface Design. *In: Proceedings of the International Workshop on Tools for Working with Guidelines*. October 7-8, 2000, Biarritz, France.

Wilkins, B. 2003. *MELD: A Pattern Supported Methodology for Visualization Design*. PhD Thesis, The University of Birmingham.

Yacoub, S. and H. Ammar. 2003 *Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Zimmer, W. 1995. Relationships between design patterns. *In:* J. O. Coplien and D. C. Schmidt (eds.). *Pattern Languages of Program Design*. New York: ACM Press. pp. 345-364.

Zimmerman, J., S. Evenson, K. Baumann and P. Purgathofer. 2004. Workshop on the Relationship between Design and HCI. *In: Proceedings of Human Factors in Computing Systems (CHI 2004)*. April 24-29, 2004, Vienna, Austria. pp. 1741-1742.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix A: Interactive Design Methods and Limitations

Scenario-based design [Carroll 2000] makes system use explicit by capturing one particular instance of user-system behavior. According to Carroll, a number of properties add to the complexity of the design process: Incomplete problem descriptions, lack of design guidelines, difficulties in predicting the design goal or solution state, and decision trade-offs. Furthermore, designers are required to have a diverse skill set, in a domain which has a wide-ranging impact on human activity. To help solve some of these problems and limit the assumptions designers make about users and tasks, *scenarios of use* can be employed. Scenarios depict *stories* about users, behaviors and actions, with the underlying premise that software transforms human activity and is constrained by the actual context of that activity.

Scenarios can be gathered using ethnography, in addition to other data collection techniques. The key is to include enough detail to make significant design issues apparent, and to include characteristics such as setting, actors, goals and actions. Examination of scenarios is then facilitated by *claim analysis*, a process which solicits application of HCI knowledge and theories. Claim analysis is paved by certain activities such as scanning for cause and effect, systematic questioning, and participatory analysis – with the purpose of clarifying design rationale, including any design decisions and trade-offs. Designs based on scenarios make *use* an important aspect of the software's resulting conceptual design. Therefore, it is alleged that scenario-based design, as an interactive design method, addresses both usability and quality in use. In addition, scenarios are flexible enough to be integrated in many design methods and development lifecycles, and can play a part in contextual design, usage-centered design and goal-directed design.

However, saying all that, we still have a long way to go from stories captured as text to concrete design solutions. Although scenario-based design aims to make system use explicit albeit through narrative description, the resulting UI conceptual model still relies on the experience and background of designers and developers. The resulting design is not based on any systematic process, nor can it be mapped directly from scenarios.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Usage-centered design [Constantine and Lockwood 1999] focuses on *usage*. Central to this method is its use of models to drive design, its focus on improved tools to support task accomplishment, and its selective user involvement. Different facets of interactive systems are abstracted into models including the user, task and content models. The user model incorporates user needs, characteristics, behaviors and expectations, with the objective of modeling the user-system relationship. The task model is comprised of task descriptions of user-system interactions, with diagrammatical representations between different tasks. Finally, the content model depicts the organized use of user interface building blocks called abstract prototypes, with a context navigation map describing the flow between them.

Supporters of this design method claim that it overcomes the main limitation of UCD – which is too much focus on users, and not enough on usage. This claim, according to its proponents, is motivated by a number of underlying reasons. First, user studies easily confuse what users want with what they actually need. Secondly, a proper design based on thoughtful and systematic progressions from the beginning is far more effective than rapid iterative prototyping. Thirdly and most controversially, usability testing can be a relatively inefficient way to find problems that could have been avoided through proper design.

Despite Usage-centered design's alleged advantages over UCD and its doctrine of "proper design", there is still little explanation on how developers are expected to arrive at such a proper design. The models generalize various properties of the user interface at different levels of abstraction and detail. "Abstract prototypes" from the content model should be based on information contained within the user and task models, however it is unclear how this progression occurs, and how designs are actually derived.

Contextual design [Holtzbatt and Beyer 1998] concentrates on integrating customer-related data into design through a set of defined activities: Contextual inquiry, work modeling, consolidation, work redesign and user environment design. Ethnographic data is first collected as part of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

contextual inquiry, with the purpose of aligning the customer and designer view of the envisioned interactive system. Work models are built as flow, sequence, artifact, cultural and physical diagrammatic representations, to detail the work of each interviewed customer. The work models are consolidated with the hopes that an underlying pattern or structure will result, demonstrating the work that the design must address. By employing techniques such as scenarios and storyboards, the team attempts to create a vision for work redesign, aimed at improving the customer's work.

To this end, as part of user environment design, work functions and objects are organized as *focus areas*. These focus areas are described with a purpose, function, links to other foci, interaction objects, constraints (such as speed), and open design issues (such as implementation concerns). It is only after this step that system mock-ups can be created, tested with customers and refined, giving way to implementation. Proponents of contextual design [Holtzblatt and Jones 1995] claim that it meets the challenge of combining innovative design with usability engineering. In addition, it is a comprehensive approach to interaction design since it embraces business, organizational, work and user experiences.

Although contextual design aims to integrate user data into the design process, there is still a fuzzy link between the captured data and the resulting design solutions. In essence, the focus areas represent possible UI constructs, but it is not well-defined how one arrives at the actual design solutions.

Participatory design [Ehn 1988] promotes interactive system design as a social and creative activity requiring the involvement of users and designers with varying backgrounds. The idea is to combine the skills and knowledge of users or representative users of the system, with the technological and organizational expertise of designers and developers. Participatory techniques include collaborative design, ethnography, participative analysis of usability data, action research, contextual inquiry, co-development and cooperative evaluation [Muller et al.1993].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In recent years, participatory design has increasingly been adopted in industry, in the context of design workshops. Typically, developers, business representatives and users work together to come up with usable design solutions. Advantages of employing such a method include giving users a voice in the design process, enabling both technical and non-technical participants to provide feedback, as well as providing a forum for developers to meet their users and identify key issues. Agendas for such workshops vary, but typically include facilitators, commence with usability guidelines, and detail objectives and expectations – all before moving onto discussions about design solutions.

Design goals, scenarios, and even paper prototypes are often discussed and constructed in participatory design workshops. Although the involvement of users and their experiences alongside of designers and developers is central to this method, once again, a clearly-defined process leading to design is lacking. User experiences are captured as narrative descriptions and incorporated into techniques such as scenarios and storyboards, however what happens after that is based solely on the opinions and experiences of the individuals that are present. One can argue that since design solutions are constructed with both users and designers present, it is a positive step in the right direction compared to other interactive design methods described in this document. Nevertheless, a systematic process is still missing, and there is little focus on reuse of best practices in design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix B: The UPADE Web Language

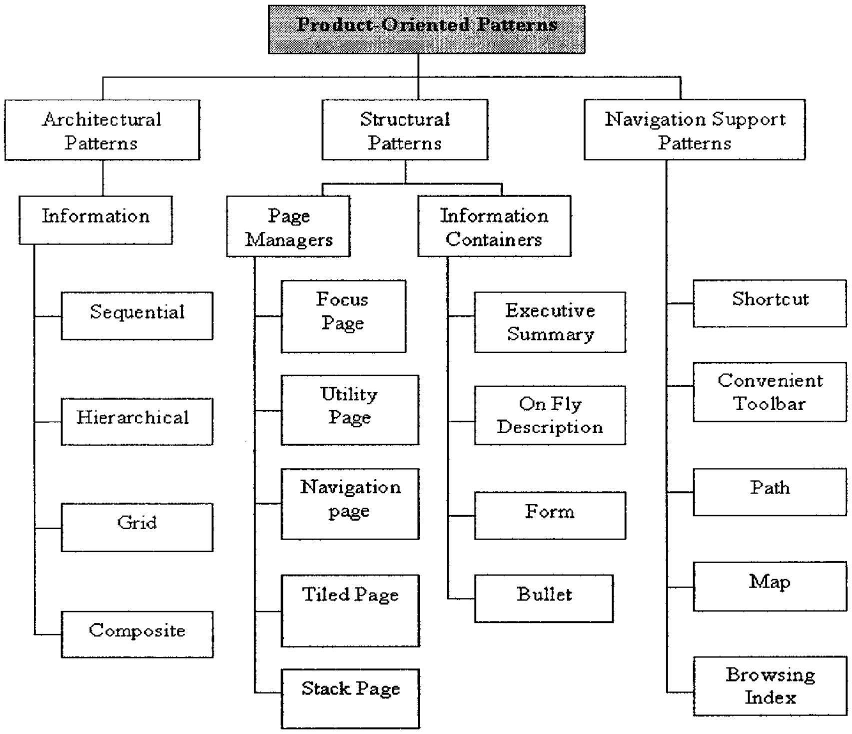


Figure B-1. An Overview of the UPADE Web Language

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix C: Pre-Design Usability Evaluation of NCBI Site

Psychometric Evaluation with End-Users

Psychometric assessment, through objective questionnaires administered to users, is a usability evaluation method where questionnaires are used to gather user perceptions in a systematic way. They are analogous to structured interviews in that questions are presented in the same way to all respondents. In the past, we have used the Software Usability Measurement Inventory (SUMI) to assess user perceptions of a software system [Kirakowski and Corbett 1993]. The SUMI is a multi-dimensional inventory questionnaire that evaluates different aspects of user satisfaction. However, for this study, we decided to administer a tailored questionnaire for two particular reasons. First, since we are dealing with a web-based information system, the mode of interaction between users and the system is different from a traditional software system, and therefore necessitates specific web-based scenarios and questions. Secondly, the study entailed participation from a particular user group and community that uses the NCBI site to accomplish specific tasks. We wanted to make sure that the questionnaire was domain-specific, and asked appropriate task-related questions.

The questionnaire created for usability evaluation with end-users consisted of three parts: (1) User Information, (2) User Evaluation of NCBI Site, and (3) General Questions. The following experts were consulted to assure both quality and precision in terms of the quantifiers used: A cognitive psychologist, a senior usability expert, and a biomedical specialist. The purpose of the first part of the questionnaire was to gather some demographic and user information (e.g. age, gender, knowledge and experience), while the third part was aimed at gathering general impressions of the site from the user (e.g. what do you like and dislike about the NCBI site's homepage?).

The second part of the questionnaire contained specific questions, which enabled us to quantify user experiences with certain properties of the site. Similar to McKenzie [2000], heuristics were

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

used to describe different facets or properties of the site. In other words, each set of questions was correlated with a particular heuristic. Using the same list of nine heuristics introduced earlier in this section, we were able to cover the most important aspects of usability by asking specific questions. For example, if we are to take the first heuristic from Table C-1 ("Visibility and Navigation"), the following questions were asked to assess user experiences with relation to the visibility and navigation of the site:

- Do you find it easy to navigate on the NCBI website, especially when performing a new task?
- Is it visually clear what is a link or a button?
- Do you receive feedback and requested information promptly, such as when you perform a BLAST search?
- Is it easy to get lost when looking for information?

Heuristic Evaluation with UI Experts

We carried out a second type of test with UI experts. In contrast to the psychometric evaluation, heuristic evaluation is an inspection method where UI experts evaluate a user interface directly against the heuristics [Nielsen 2001]. Early lists of heuristics were lengthy and difficult to apply. To reduce testing costs, Nielsen came up with a list of ten heuristics that cover what he considered the most important aspects of usability. We adapted this list to a set of nine heuristics for the web. A subset is illustrated in Table C-1. Nielsen [2001] claims as few as 3-5 experienced evaluators are necessary for heuristic evaluation, stating that they will be able to detect the majority of usability problems. Since heuristic evaluation is a subjective usability method, it has been used as an approach for building first a qualitative picture of user experiences. To a certain extent, such a picture was used to tailor the questionnaire we conducted with end-users.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Heuristic	Definition
1. Visibility and Navigation	Sections and links should be clearly marked, and users need to know "Where am I?" and "Where can I go next?" System should keep users informed about what is going on, through feedback within reasonable time (e.g. progress indicators).
2. Language and Communication	The system should "speak" in a language familiar to the user (e.g. technical terms should be avoided).
3. Control	The user should always feel like they have a way out of the system and unwanted states, such as with the "home" button

Table C-1: Subset of Heuristics [adapted from Nielsen 1994]

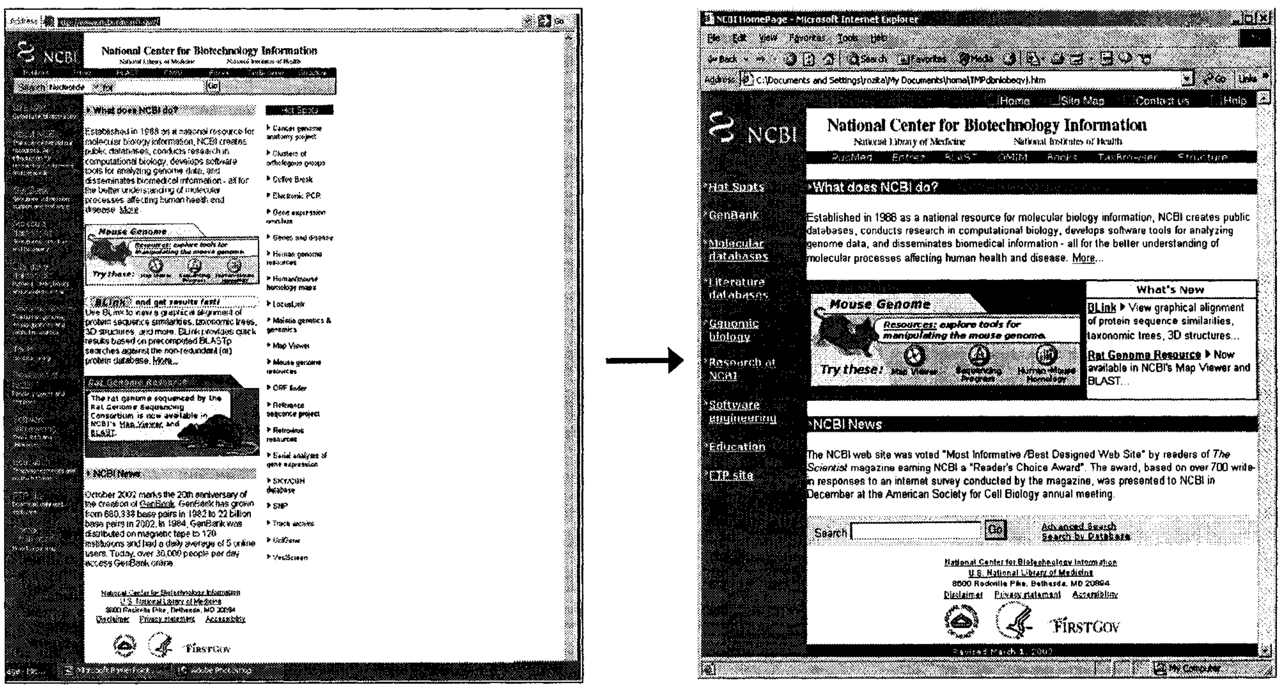


Figure C-1. Original and redesigned NCBI homepage using patterns

Design Decisions

Some examples of design decisions (for the homepage) based on personas, usability evaluation results and existing pattern languages were:

- Navigation menu on the left has important site links, however a textual description of the link

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

is provided under each name; this is only useful for novice users who don't know what kind of information they can find from the links. This convenience for the novice user is a hindrance for an expert user as it contributes to more scrolling. A solution would be implementation as rollovers (on-fly descriptions) to help new users.

- Hot Spots are representative of possible shortcut links that are only useful for expert users who know what they are looking for; these can be updated and changed with time.
- General information about NCBI, what they do, and their mandate are interesting for novice users; for expert users, this clutters the site.
- NCBI news and newsletter is a good idea for expert users and should be detailed on the homepage for this group; it may be a good idea to place it instead of the general NCBI information. However, for novice users, it will just add additional content and scroll down. A good compromise is to replace it with a banner link, and place the content on another page.
- The explanation of the three information containers on the page are useful for novice users, but may just be adding extra content for expert users. More usability studies would need to confirm this since the containers are not static. Saying that, it may still be useful to have the existing links, but without such detailed explanations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix D: Task-Based Evaluations

A. NCBI Study

Task 1

Scenario:
You are a Bioinformatics researcher working at the BC Cancer Agency. At a recent convention, you met Stephen Bryant, a senior investigator at NCBI. You now need to look up what he does for NCBI.

Method:
For purposes of this exercise, you will be asked to verbalize each step performed as you are performing the task.

Task Steps:
1. Start at the NCBI Homepage.
2. Look for the projects Stephen Bryant manages at NCBI.
3. Return to the NCBI Homepage.

Task 2

Scenario:
You are a graduate student in Human Genetics. You have obtained some new protein sequences, and you need to view their 3-dimensional structure using the Cn3D utility.

Method:
For purposes of this exercise, you will be asked to verbalize each step performed as you are performing the task.

Task Steps:
1. Start at the NCBI Homepage.
2. Find the link that allows you to download the Cn3D utility.
3. Stay on the current page.

Task 3

Scenario:
You have been trying to perform a nucleotide-protein search, but you keep receiving an error with your query. You realize that you need technical assistance.

Method:
For purposes of this exercise, you will be asked to verbalize each step performed as you are performing the task.

Task Steps:

1. Start from where you are currently on the page.
2. Find out to whom to report a technical support question for this scenario.
3. Return to the NCBI Homepage.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Task 4

Scenario:
You are a Genetics graduate student. You want to improve your knowledge of the NCBI site and offerings, so you have decided to take the Field Guide Plus course, an enhanced NCBI training course.

Method:
For purposes of this exercise, you will be asked to verbalize each step performed as you are performing the task.

Task Steps:
1. Start at the NCBI Homepage.
2. Find the information page detailing how to register for this course.
3. Return to the NCBI Homepage.

B. Protein Explorer Study

Task 1: Exploring the Hemoglobin molecule

1. Load Hemoglobin structure, with the PDB code 1HGA.
2. Stop the molecule from spinning.
3. Remove the ligands from the molecule.
4. Modify view to "spacefill" (from "backbone") for all atoms of the molecule.
5. Find out more about this molecule. For example, title and taxonomic source.
6. In advanced options, find out how to view surfaces (multiple surfaces concurrently). This will allow you to use options such as "molecular electrostatic potential".

Task 2: Exploring the Insulin molecule

1. Load Insulin structure, with the PDB code 1APH.
2. Zoom into the molecule.
3. Remove water from the molecule.
4. Modify view by changing the color scheme to "black" for "chain A" of the molecule.
5. Find out more about this molecule. Specifically, information about reliability of the model.
6. In advanced options, find out how to view surfaces (multiple surfaces concurrently). This will allow you to use options such as "molecular electrostatic potential".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix E: Patterns for Low Literacy & Cognitive Disability

Context

The examples included in the patterns below are part of the Montreal City website, at http://ville.montreal.qc.ca. As an outcome of a Montreal summit in 2002, the city began a universal access initiative for the disabled. More than 30% of the population of Montreal has various reasons for difficulty with reading. In short, universal access allows individuals with limitations and disabilities (visual, auditory, intellectual or others), to participate in the community and to engage in the autonomous use of products and services. Therefore, a sub-section of the site has been constructed for people with intellectual disabilities or major reading and language problems. The City of Montreal was the first municipality worldwide to construct this kind of a web site, adapted for citizens with disabilities.

We have used the site as a guide to construct three patterns which can be reused for individuals with a low literacy level, and/or with cognitive disabilities. The work is based on guidelines referenced in [Montreal City 2006].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Pattern 1: Simplified Text

The *Simplified Text* pattern offers information in a condensed, as well as easy to read and understand format.

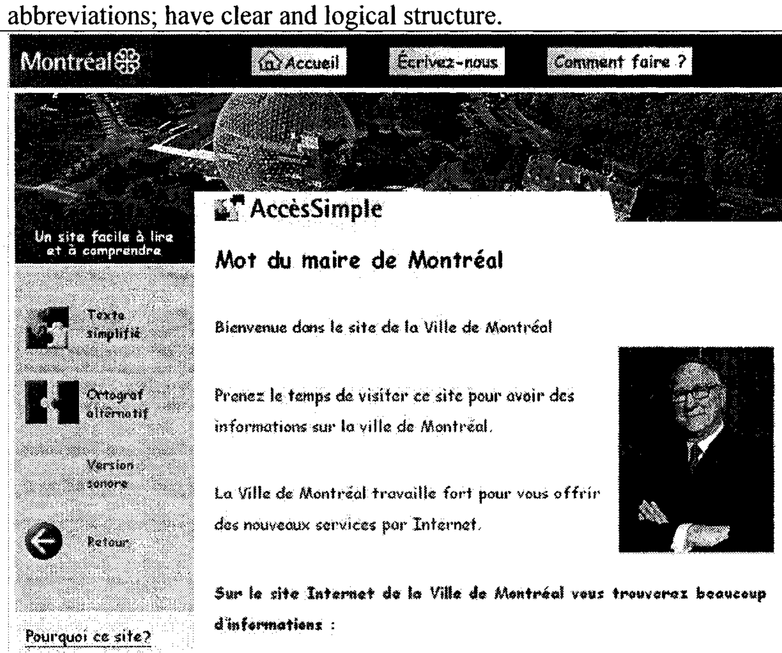
Title	Simplified text
Context	The user has difficulties in reading and understanding information which is read.
Problem	How to reduce the complexity of the communicated information without necessarily impeding the content
Forces	need easy to understand content need simple organization of content
Solution	Offer information in a condensed easy to read and understand form. Use simple and direct language; present a single main idea in a phrase; avoid abbreviations; have clear and logical structure.
Examples	 from [http://ville.montreal.qc.ca]
Related Patterns	Neighboring: Alternative Orthography, Audio Communication

Table E-1. Simplified Text Pattern

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Pattern 2: Alternative Orthography

Further simplification of text is required for a certain portion of the population. In addition to simplifying the content, the *Alternative Writing* pattern reduces the complexity of the writing. This writing style is based on phonetic spelling. Alternative writing uses only 35 relations whereas the conventional French language uses more then 4000. It should be noted that alternative writing is not a new way to write in French. It is a different way of presenting the language, similar to how brail is used for the blind.

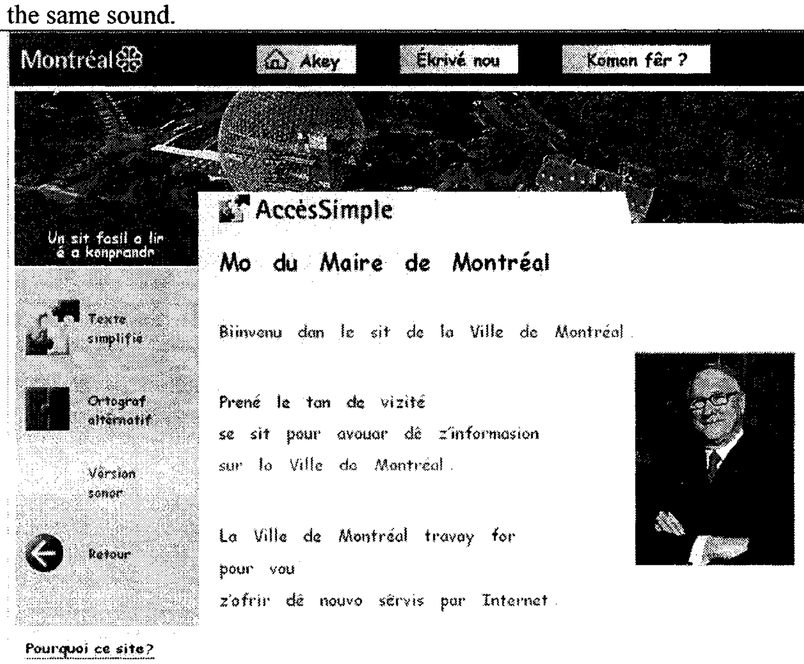
Title	Alternative Orthography
Context	The user has serious difficulties in reading and understanding information because of limited intellectual capacity or low literacy.
Problem	How to reduce the complexity of the medium used (language) to communicate information without necessarily impeding the content.
Forces	may require simplified written communication; may require spelling phonetically
Solution	Reduce the complexity of the written language by simplifying the orthography. Construct an alternative writing style with a simple orthography with a goal to allow for "write it as you hear it". Use only one combination of letters to produce the same sound.
Examples	 from [http://ville.montreal.qc.ca] • Simplifies French orthography by reducing the amount of relations between letters from 4000 to 35.
Related Patterns	Neighboring: Simplified Text, Audio Communication.

Table E-2: Alternative Orthography Pattern

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Pattern 3: Audio Communication

The *Audio Communication* pattern offers a complementary means to communicate all the text presented in the previously described formats. This is particularly important for individuals which are illiterate.

Title	Audio Communication
Context	The user is not capable of reading and understanding information because of limited intellectual capacity or illiteracy.
Problem	How to present information in a format understood by an illiterate user.
Forces	Communicates information in a simple and well-organized format. Does not require any knowledge of written communication.
Solution	Allow for audio communication of the information presented on the screen. The information should be organized and presented in audio format allowing for the easy understanding and assimilation of information. When possible, provide a means for audio interaction between system and user.
Examples	
Related Patterns	Neighboring: Simplified Text, Alternative Orthography.

Table E-3: Audio Communication Pattern

The figures below compare the homepage of the two different sites; the first one for the general public and the second one for universal access [Montreal City 2006].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Figure E-1: Homepage for general public

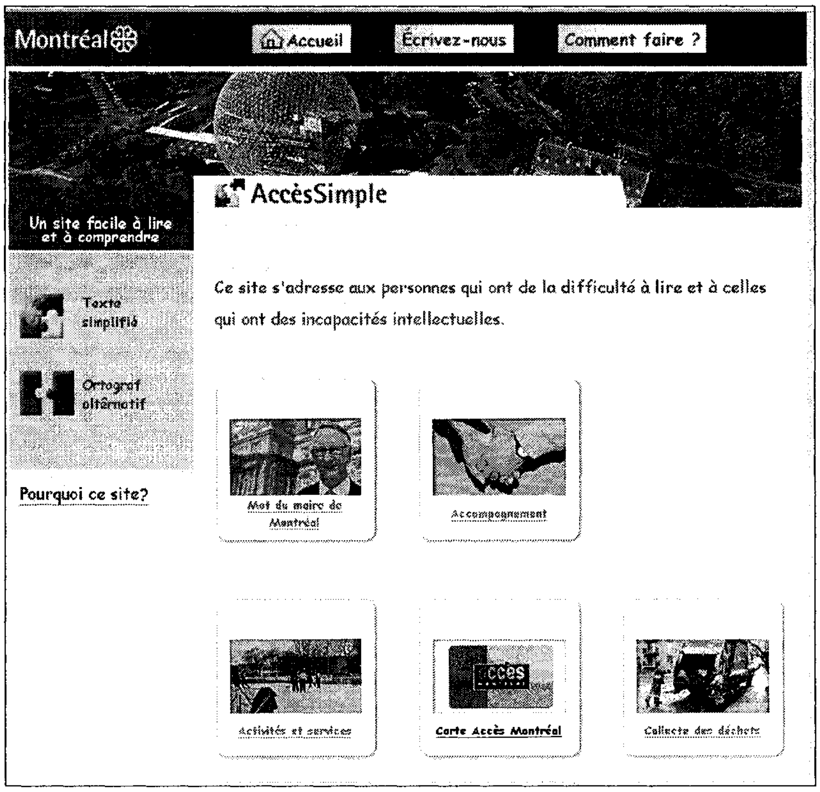


Figure E-2: Homepage for universal access

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix F: Rule-based scoring technique

<table>
<tr><th>Variable</th><th>Pseudo code</th><th>Code</th></tr>
<tr><td>Linguistic Ability</td><td>As linguistic ability decreases
NM increases proportionally</td><td>if (ling_ability >= 0)
then NM += max(ling_ability)-ling_ability;</td></tr>
<tr><td>Literacy</td><td>When literacy is 0 (illiterate)
S, SL, NM, CL increase by 4 pts
When literacy is 1 (low)
S, SL, NM, CL increase by 3 pts</td><td>if (literacy == 0 || literacy == 1
then
{ S += 4-literacy;
SL += 4-literacy;
NM += 4-literacy;
CL += 4-literacy; }</td></tr>
<tr><td>Computer Experience</td><td>When computer exp is 0 (none)
G, S, SL, NM increase by 4 pts
When computer exp is 1 (low)
G, S, SL, NM increase by 3 pts</td><td>if (comp_exp = = 0 || comp_exp = = 1)
then
{ G += 4 - comp_exp;
S += 4 - comp_exp;
SL += 4 - comp_exp;
NM += 4-comp_exp; }</td></tr>
<tr><td>Education Level</td><td>As education level decreases
SL, CL, G increase proportionally</td><td>if (educ_level >= 0)
then
{ SL += max_educ_level- educ_elevel;
CL += max_educ_level - educ_level;
G += max_educ_level - educ_level; }</td></tr>
<tr><td>Product Experience</td><td>When product exp is 0 (none)
G, S, SL, NM increase by 4 pts
When product exp is 1 (low)
G, S, SL, NM increase by 3 pts
When product exp is 3 (high)
E, C increase by 3 pts
When product exp is 4 (expert)
E, C increase by 4 pts</td><td>if (app_exp = =0 || app_exp = = 1)
then
{ G += 4- app_exp;
S += 4- app_exp;
SL += 4- app_exp;
NM += 4- app_exp; }
else if (app_exp = = 3 || app_exp = = 4)
then
{ E += app_exp;
C += app_exp; }</td></tr>
<tr><td>Age</td><td>When age is 0 (toddler)
A increases by 4 pts
SL, CL increase by 3 pts
When age is 1 (child)
A increases by 3 pts
SL, CL increases by 2 pts
When age is 2 (adolescent)
A increases by 2 pts
When age is 5 (senior)
SL, G, S, CL increase by 3 pts
When age is 6 (elderly)
SL, G, S, CL increase by 4 pts</td><td>if (age == 0 || age == 1)
then
{ SL += 3-age; CL +=3 -age; A +=4 -age; }
else if (age = = 2)
then
{ A += 2; }
else if (age = = 5 || age = = 6)
then
{ SL += age - 2; G += age - 2; S += age - 2;
CL += age - 2; }</td></tr>
<tr><td>Need for guidance</td><td>As need for guidance increases
G increases proportionally</td><td>if (need_guidance >= 0)
then G += need_guidance;</td></tr>
<tr><td>Need for validation of decisions</td><td>As need for validation of decisions increases
G and S increase proportionally</td><td>if (need_val_decisions >= 0)
then
{ G += need_val_decisions;
S += need_val_decisions; }</td></tr>
<tr><td>Need for Control</td><td>As need for control increases
C increases proportionally</td><td>if (need_control >= 0)
then C += need_control;</td></tr>
<tr><td>Behavior towards features</td><td>As behavior towards features decreases
CL, S, SL increases proportionally</td><td>if (feature >= 0)
then
{ CL += max_feature) - feature;
S += max_feature) - feature;
SL += max_feature) - feature; }</td></tr>
<tr><td>Initiative taking</td><td>As initiative taking decreases
G increases proportionally
As initiative taking increases
G increases proportionally</td><td>if (initiative >= 0)
then
{ G += max_initiative) - initiative;
C += initiative; }</td></tr>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Variable	Pseudo code	Code

IT attitude	As IT attitude decreases
	A, G, CL, S increase proportionally
		`if (it_attitude >= 0)`
		`then`
		`{ A += max_it_attitude) - attitude;`
		`G += max_it_attitude) - attitude;`
		`CL += max_it_attitude) - attitude;`
		`S += max_it_attitude) - attitude; }`

Level of motivation	As level of motivation decreases A increases proportionally	`if (motivation >= 0)` `then A += max_motivation) - motivation;`

Legend: Appeal (A); Min. Cognitive Load (CL); Control (C); Efficiency in Use (E); Guidance (G); Natural Mapping (NM); Safety (S); Simplicity (SL)

Table F-1: Rules for associating user variables to usability needs

User Need	Pattern Criteria
Appeal	Logical Organization, Minimalist Design
Control	Consistency, Feedback, Facilitated Navigation
Efficiency of Use	Accelerators, Logical Organization, Minimalist Design
Guidance	Feedback, Facilitated Navigation
Min. Cognitive Load	Consistency, Logical Organization, Minimalist Design, Facilitated Navigation
Natural Mapping	Consistency, Facilitated Navigation
Safety	Error Handling, Error Prevention, Feedback
Simplicity	Consistency, Minimalist Design

Table F-2: Associating user needs to pattern criteria

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix G: Persona Example for Protein Explorer

Persona 1: Marta Aviles		**"I want to be able to access specific features without having to worry about the details"**
Identifier	Young adult, medium domain experience, computer science background	
General Profile	Marta is a 27 year-old computational biologist in a pharmaceutical company. She has a Masters in Computer Science. She is a professional woman working in a highly-competitive industrial setting, and works over 50 hours/week. She lives alone and likes to wine and dine with friends in her spare time. She takes her work seriously, and would like to move up to middle-management eventually. She uses various Bioinformatics tools on a daily basis, working mostly with predictive models for RNA. She uses 3D visualization features to assess the predictive power of specific molecular interactions. She needs to manipulate the visualization easily, and see the inner workings of the tool (computations). She is very comfortable with computing technology, and has experience programming in database and object-oriented languages. She is often in a rush, and is always struggling to keep up with deadlines. She doesn't like to stay late at work, but she typically ends up doing overtime.	
Goals	Professional: Move up to middle-management. Personal: Get married and settle down. Application: Works with predictive models and uses advanced 3D visualization features.	
Scenario	Description	Marta is working on a new predictive model for determining protein structure. She just read a paper on a new method, and wants to examine some ideas she has with her work on the Hemoglobin molecule. She is sitting in her office, munching away on her lunch, and interacting with the 3D visualization tool. She searches for the option to view multiple surfaces concurrently, but has some trouble setting the initial parameters for determining molecular electrostatic potential. She only needs access to this feature, and is not interested in setting other constraints on the molecule. She gets slightly confused with all the advanced biomedical terminology, but after spending 20 minutes trying different values, she gets one that she is happy with.
	Specific needs	Control and behavior to features (average). Balance between efficiency and simplicity.
	Features	Advanced options, customizing parameters, tracing inner workings of tool
	Interaction details	Marta does not give up easily if she can't manipulate the molecule in the way she would like or if she can't find the information she needs. She wants to get the job done, without worrying about all the details. She is comfortable with computing technology and terminology.
Demographics	Marta is a 27-year old female, with an average family income. She has no dependents.	
Knowledge and Experience	Marta is fluent in English and Spanish. She has a high education level (Master's in Computer Science), and has average domain experience in Bioinformatics. She is an expert computer user, and has advanced programming experience.	

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Psychological profile and needs	Marta has little need for validation of decisions and guidance. She is a proactive user, and has somewhat of a high learning speed. She has an average need for control and behavior to features.
Attitude and Motivation	Marta has a positive attitude to IT, and a somewhat high level of motivation to use the system.
Special Needs	Marta has no disabilities but acts at times like the expert user group.

Table G-1: Persona 1 for Protein Explorer

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix H: Empirical Studies in HCI

In HCI and empirical software engineering, user-oriented studies are required both to motivate the research as well to assess the validation and accuracy of the proposals. In this thesis, we designed our studies according to: (1) Experimental assessment techniques from software engineering, and (2) Usability testing methods from HCI.

In software engineering [Fenton and Pfleeger 1998], three main types of experimental assessment techniques exist to evaluate a technique, method or tool: (1) surveys, (2) case studies, and (3) formal experiments. **Surveys** are retrospective studies, aiming to document relationships and trends after a particular event has occurred. Variables cannot be manipulated in this type of study, and large groups of projects and data sets are analyzed. **Case studies** are planned experiments where factors affecting a particular outcome are identified and documented. A typical project or situation is followed and analyzed, but the goal is not to capture information about all possible cases. In contrast to case studies, **formal experiments** are rigorously controlled investigations of a particular activity. Key factors are identified and manipulated, and the resulting outcome is documented. Since a great deal of control is required for formal investigations, the project scale is usually small, involving only a small set of people and events.

The main guidelines used to determine which experimental technique is suitable for a particular investigation are whether a study is retrospective and the amount of control needed (or possible). In this thesis, two large empirical studies were carried out with end-users to test and validate the UX-P process. Since our study was not retrospective and the difficulty of control was high, case studies were the logical choice [Fenton and Pfleeger 1998]. The following are the planning steps for a case study:

- Conception, where the objectives and goals of the study are clearly stated, to allow for evaluation.
- Design, where the objectives are translated into a formal hypothesis and components of the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

experiment are clearly defined. The experiment is formally defined as a set of trials, where a trial is an individual test.

  - Null hypothesis: No significant difference exists between the two treatments
  - Alternative hypothesis: A significant difference exists between the two treatments
  - Treatment: The application of a method or tool for evaluation
  - Dependent variables: Factors that are expected to change due to applying treatment
  - Independent variables: Factors influencing the application of the treatment

- Preparation, where subjects are prepared for application of the treatment. Examples include training staff and writing out instructions.
- Execution, where the experiment is executed as described in the design.
- Analysis, where all measurements are reviewed to ensure validity and where data is analyzed with statistically.
- Decision making, where a conclusion is drawn based on the analysis results.

Each case study in this thesis, including the planning steps, is described in detail in chapter 3 (Study 1) and in chapter 7 (Study 2). The tables below provide a summary according to the steps outlined above.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Study 1: NCBI	
Step	**Description**
Conception	▪ Experimental Technique: Case study ▪ Objectives: Evaluate whether framework results in more usable systems; evaluate validity of correlating personas and patterns while designing; and understand limitations of framework. ▪ Overall goal: The above would lead us to conclude that we can either develop a more concrete method and process, or that our framework needs to be refined.
Design	▪ Null hypothesis: There is no relationship between employment of the proposed framework and more usable designs satisfying the collection of user experiences. ▪ Alternative hypothesis: The proposed framework will lead to designs that can be argued as more usable designs satisfying the collection of user experiences. ▪ Treatment: Application of the Persona to Pattern Framework (chapter 3). ▪ Dependent variables: Task duration, failure rates, satisfaction ratings. ▪ Independent variable: Design type (Original or New).
Preparation	▪ 39 end-users participated; 16 for pre-design and 23 for post-design. ▪ 3 UI experts participated in heuristic evaluation. ▪ 2 UI experts carried out application of framework and implemented the *New* design.
Execution	The two treatments were tested using usability testing methods and a *between-subjects protocol* where users were randomized into two treatment groups based on design type.
Analysis	▪ Task duration: Overall improvement of 55% (statistically significant) for *New* design. ▪ Failure rates: Overall failure rates decreased (statistically significant) for *New* design. ▪ Satisfaction ratings: Satisfaction higher (statistically significant) for *New* design. ▪ Based on the above, the null hypothesis is refuted.
Decision making	Main conclusion is that the development of a more concrete method and process based on the framework is founded. ▪ Usability indicators (task duration, failure rates, satisfaction ratings) indicated an improvement in terms of usability for the New design compared to the Original design. ▪ Application of the framework was found to facilitate the design activity, allowing the incorporation of sound UCD principles and providing structure. ▪ Limitations to be addressed: Further development of framework should include identifiable and discrete steps, formalization of persona and pattern information, and enhancement of persona descriptions.

Table H-1: Experimental Assessment Summary for NCBI Study

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Study 2: Protein Explorer	
Step	**Description**
Conception	▪ Experimental Technique: Case study ▪ Objectives: (a) Assess the applicability of using the UX-P Process and P2P Mapper tool within the context of a HCI project, and (b) to evaluate whether the UX-P process leads to more usable systems. ▪ Overall goal: Carry out an initial validation study for the UX-P Process within the context of a real project.
Design	▪ Null hypothesis: There is no relationship between employment of the proposed process and more usable designs satisfying the collection of user experiences. ▪ Alternative hypothesis: The proposed process will lead to designs that can be argued as more usable designs satisfying the collection of user experiences. ▪ Treatment: Application of the UX-P Process (chapter 6). ▪ Dependent variables: Task duration, failure rates, satisfaction ratings. ▪ Independent variables: Design type (Original or New), variation of design order
Preparation	▪ 22 end-users participated; 22 for pre-design and a subset of 15 for post-design. ▪ 2 UI experts carried out application of UX-P Process and implemented the *New* design.
Execution	▪ The two treatments were tested using usability testing methods and a *within-subjects protocol* where users perform under each condition. The design order was varied per user to reduce learning effects.
Analysis	▪ Task duration: Overall improvement of 52% (statistically significant) for *New* design; no significant effect from design order ▪ Failure rates: Overall failure rates decreased (statistically significant) for *New* design; no significant effect from design order ▪ Satisfaction ratings: These ratings were qualitatively assessed, with 13 out of 15 users indicating a higher satisfaction higher for *New* design. ▪ Based on the above, the null hypothesis is refuted.
Decision making	Main conclusion is that the UX-P Process led to a more usable design. Furthermore, the process was effectively applied within a HCI context and with tool support. ▪ Usability indicators (task duration, failure rates, satisfaction ratings) indicated an improvement in terms of usability for the New design compared to the Original design. ▪ Application of the process was found to facilitate the design activity, allowing the incorporation of sound UCD principles and providing structure.

Table H-2: Experimental Assessment Summary for Protein Explorer Study

The usability testing methods used for both case studies were based on well-established methods from the HCI community, testing both qualitative and quantitative factors, and reported according to the Common Industry Format (CIF) [ISO 2006]. In large part, testing methods in Human-Computer Interaction (HCI) draw foundations and expertise from psychology experimentation.

Testing, Inspection, and Inquiry methods [Ivory and Hearst 2001] were used in our case studies.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

First, *Testing* methods consist of an evaluator observing users interacting with an interface, usually while they are completing tasks, to determine usability problems and compile usage data.

Method Technique	Description
Testing	
Thinking-Aloud Protocol *	user talks during test
Question-Asking Protocol	tester asks user questions
Shadowing Method	expert explains user actions to tester
Coaching Method *	user can ask an expert questions
Teaching Method	expert user teaches novice user
Codiscovery Learning	two users collaborate
Performance Measurement *	tester records usage data during test
Log File Analysis *	tester analyzes usage data
Retrospective Testing	tester reviews videotape with user
Remote Testing	tester and user are not colocated during test
Inspection	
Guideline Review	expert checks guideline conformance
Cognitive Walkthrough	expert simulates user's problem solving
Pluralistic Walkthrough	multiple people conduct cognitive walkthrough
Heuristic Evaluation *	expert identifies violations of heuristics
Perspective-Based Inspection	expert conducts narrowly focused heuristic evaluation
Feature Inspection	expert evaluates product features
Formal Usability Inspection	expert conducts formal heuristic evaluation
Consistency Inspection	expert checks consistency across products
Standards Inspection	expert checks for standards compliance
Inquiry	
Contextual Inquiry *	interviewer questions users in their environment
Field Observation *	interviewer observes system use in user's environment
Focus Groups	multiple users participate in a discussion session
Interviews *	one user participates in a discussion session
Surveys	interviewer asks user specific questions
Questionnaires *	user provides answers to specific questions
Self-Reporting Logs	user records UI operations
Screen Snapshots	user captures UI screens
User Feedback	user submits comments
Analytical Modeling	
GOMS Analysis	predict execution and learning time
UIDE Analysis	conduct GOMS analysis within a UIDE
Cognitive Task Analysis	predict usability problems
Task-Environment Analysis	assess mapping of user's goals into UI tasks
Knowledge Analysis	predict learnability
Design Analysis	assess design complexity
Programmable User Models	write program that acts like a user
Simulation	
Information Proc. Modeling	mimic user interaction
Petri Net Modeling	mimic user interaction from usage data
Genetic Algorithm Modeling	mimic novice user interaction
Information Scent Modeling	mimic Web site navigation

Table H-3: Usability Evaluation Methods [Ivory and Hearst 2001]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Secondly, *Inspection* methods consist of an evaluator identifying potential usability problems with an application, by using a set of heuristics. Thirdly, in *Inquiry* methods, feedback is gathered from users about the application via interviews, surveys, etc. Table H-3 lists and defines the techniques applicable for each method; an asterisk (*) is beside the techniques used. The specific sequence and testing protocol for each study can be found in chapters 3 and 7.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix I: Process Described in Pseudo-code

<u>**UXP Process**</u>

```
//Persona Creation
REPEAT
    CALL cluster_users WITH user_info RETURNING cluster_set
UNTIL (cluster_set fits context of use)
Refine persona_set based on steps defined on [p.112-113]

//Pattern Selection
REPEAT
    CALL select_patterns WITH pattern_library, persona_set RETURNING pattern_set
UNTIL (pattern_set satisfies designer)
CALL filter WITH pattern_set RETURNING filtered_pattern_set

//Pattern Composition
REPEAT
    CALL compose WITH filtered_pattern_set RETURNING conceptual_design
UNTIL (conceptual_design meets requirements)

RETURN conceptual_design
STOP
```

<u>**cluster_users**</u> WITH *user_info*

```
select influential_user_variables based on user_info
update influential_user_variables by selecting a subset
prioritize influential_user_variables

cluster_set= influential_user_variables

FOR each influential_user_variable in influential_user_variables
    Restrict domain of influential_user_variable
END LOOP

cluster_set = cross product of values of influential_user_variables
FOR each cluster in cluster_set
        Examine applicability of cluster based on criteria defined on [p.106-107]
END LOOP
RETURN cluster_set
STOP
```

<u>**select_patterns**</u> WITH *pattern_library, persona_set*

```
FOR each persona in persona_set
        Perform steps defined on [p.119-120]
END LOOP
```


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

RETURN *pattern_set*
STOP

---

filter WITH *pattern_set*
FOR each *pattern* in *pattern_set*
Examine applicability of *pattern* based on criteria defined on [p.128]
END LOOP
RETURN *filtered_pattern_set*
STOP

---

compose WITH *filtered_pattern_set*
construct *conceptual_design* based on steps defined on [p.129-140]
RETURN *conceptual_design*
STOP

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.