P2P Framework: A Process and a Supporting Tool for Deriving a Conceptual Design from User Experiences

Alexander Deichman

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master in Computer Science at
Concordia University
Montreal, Quebec
August 2007

Your file Votre référence
ISBN: 978-0-494-45288-2
Our file Notre référence
ISBN: 978-0-494-45288-2

ABSTRACT

P2P Framework: A Process and a Supporting Tool for Deriving a Conceptual Design from User Experiences

Alexander Deichman

Currently proposed User-Centered Design (UCD) artefacts such as patterns and personas lack commonly defined standards and tight integration into the design process. This is particularly important in the case of rapidly evolving industrial projects where high pressure from the market competes with the usability objectives. As a result, final design is often based on expert experiences without any ability to trace back the decisions.

We propose a UI design process and supporting software tightly integrating both artefacts with a goal to produce a conceptual design from user experiences. The proposed process consists of three phases: Persona Creating, Pattern Selection and Pattern Composition, resulting in representative persona set, ordered list of suggested patterns for each persona and a conceptual design respectively. On the other hand, the proposed tool supports the first two phases of the process by providing various automation algorithms for user grouping and pattern selection combined with the benefit of rapid pattern and user information access. Presented persona and pattern formats are augmented with a set of discrete domain variables to facilitate automation and provide an alternative view on the information.

Finally, the tool and the method are evaluated in the scope of a study on a bio-informatics tool. The results of the study are presented demonstrating a significant improvement in usability measures.

ACKNOWLEDGMENTS

The help and support of countless people have made this thesis a reality. A special "thank you" is due to:

Dr. Ahmed Seffah for his patience and support despite the distance and time constraints that, at times, have separated us. He gave me the opportunity to discover a new world of HCI and participate in a novel and challenging project.

Dr. Radhakrishnan, for his short but sharp comments and countless suggestions and ideas that made this thesis a reality.

Homa Javahery, a coworker who became a close friend, practically lived with me in a lab, and pushed and inspired me during the entire process.

My love, Julia, who was able to manage all my ups and downs and helped me to pass through most difficult moments of my life. She was there for me when I needed her the most.

Above all, my parents, who did everything they could to shield me from the problems of the outside world. Their strong beliefs in my abilities and their patience have been an endless source of energy for me.

Table of Contents

List of Figures

List of Tables

1. Introduction

In this chapter, we will highlight the problem of bridging the current "gap" between user interface requirements represented as personas and conceptual design, which we see as a composition of patterns. In addition, we will present an overview of the thesis structure.

1.1 Motivation

Human Computer Interaction (HCI) is often defined as a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them [Hewett et al. 1996]. It is a multidisciplinary field that has emerged in the past 20 years from an ongoing effort to minimize the barrier between human's cognitive model and an interactive model provided by the machine's communication interface. This field is concerned with techniques, methodologies, processes and tools for design, implementation and evaluation of User Interfaces (UI). Most techniques used in HCI concentrate on constant communication and feedback between users and designers in order to place a user at the center of development and develop a system to satisfy user needs, rather than forcing a user to adapt to a complete system.

These techniques provide guidelines and tools necessary for developing and documenting the design at a particular step of evolution. Personas, scenarios, task analysis, workflow modeling, context analysis etc. concentrate on analysing, modeling and capturing user experiences, tasks and context of use, mostly as narrative text. Design guidelines, principles, style sheets, patterns etc. help the designer to build a prototype, a conceptual or a detailed design. However, the development of design solutions from available analysis is ambiguous and based on guidelines and principles rather then on a systematic design. Often the final design heavily depends on the designer's expertise and background [Preece et al., 2002]. As a result, we notice a significant luck of tools and techniques that will allow the designer to link the analysis and design phases and improve traceability.

1.2 Objectives and Scope of the Thesis

In the previous section, we have noticed the lack of techniques that integrate both user experience analysis and design activities. As we believe, lack of clear specifications explaining and relating particular user experiences with the decisions taken at design level is the primary cause. Currently, there is no way to systematically derive a concrete design from user experiences [Javahery 2007]. In fact, there is no way to trace back a large part of the decisions that are made during the design and thus reproducibility of the process is compromised.

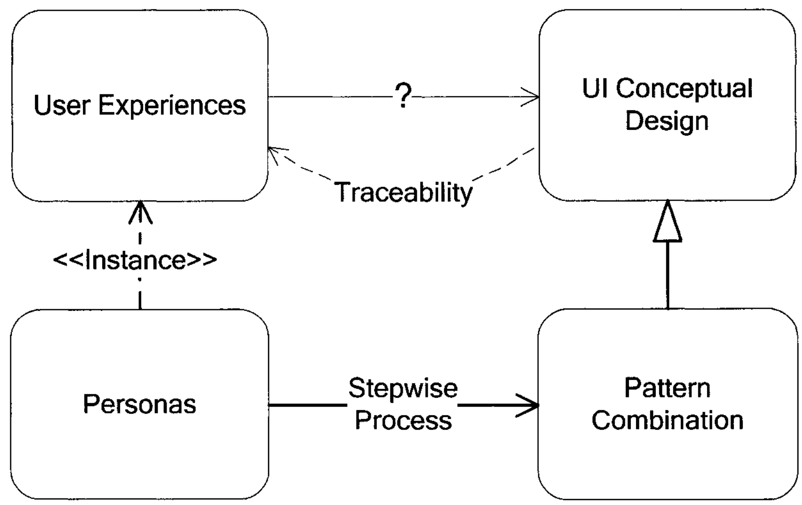


Figure 1: Deriving a Design from User Experiences

We believe that these limitations can be overcome by investigating the relations between personas and patterns (see Figure 1). In fact, we have described both as commonly used techniques to respectively capture user experiences and disseminate best practices in design. Thus, by defining a stepwise process with clearly described information flow and decision points, we can give the designers the ability to make a more enlightened traceable design decisions and learn from previous experiences based on concrete information from comparable projects and non-anecdotal experiences.

Our objectives are:

(1) To construct or adapt one of the existing UCD variations in order to clearly define all the steps bridging the gap between user experiences and a concrete design

(2) To adopt the process and all artefacts used in order to facilitate automation.

(3) Define and implement a support environment that will reduce the burden placed on the user and still follow the process constructed in (1) and adopted in (2).

1.3 Organization of the Thesis

Chapter 2 describes major challenges in use of both persona and patterns as part of UCD process and a set of requirements for the future evolution of both artifacts and UCD processes in order to solve some of these issues.

Chapter 3 describes our research approach. General requirements for process and related artifacts are presented. Our approach to information analysis, development and testing of the proposed solutions are briefly described.

Chapter 4 describes a preliminary study on bioinformatics website. Design process and results are presented in the scope of the research for a process translating user experiences into a conceptual design.

Chapter 5 presents knowledge accumulated through various studies and other elicitation activities. User and pattern formats and all acquired knowledge relating both artifacts are summarized and explained.

Chapter 6 presents, Persona to Pattern (P2P) process as one of the possible implementations of acquired expert knowledge and previously defined requirements. Process overview, the same as user, persona and pattern models are presented.

Chapter 7 details each phase of the P2P Process. Its application is demonstrated through a perspective of a fictitious case study aimed at designing a game for a mobile device.

Chapter 8 presents a proposed semi automated environment. Problems, solutions, simplifications, shortcomings and abstractions from the proposed methodology are described.

Chapter 9 describes a case study and other methods used in order to validate the process and the supporting tool. The process and the tool are evaluated in a scope of User Interface design of a protein visualization environment.

Finally, **Chapter 10** provides a summary of the work and presents possible avenues for the future research.

2. Background and Related Work

In this chapter, we will consider patterns as the medium to disseminate best practice in HCI and persona as one of the techniques commonly used to capture requirements and drive a design. Finally, we will consider a set of high-level requirements extracted from the analysis of the current use of personas and patterns.

2.1 User Centered Design: methods and techniques

From the ongoing effort to better satisfy user needs in the scope of HCI, has emerged a process and a philosophy commonly used and accepted today: User Centered Design (UCD). It is a philosophy that places a person (as opposed to a 'thing') at a center of the design; it is a process that focuses on cognitive factors (such as perception, memory, learning, problem solving, etc.) as they come into play during peoples' interactions with things [Katz-Haas 1998]. UCD has won a greatest recognition when it became part of ISO standard: ISO 13407 and ISO 9241-11. ISO 13407, Human-centered design processes for interactive systems, is a standard that provides guidance for user-centered design. This standard relies on a definition of Usability as described in ISO 9241-11. After ISO, came a large interest from industry. Although interest does not mean integration, recent studies [Venturi and Troost, 2004] demonstrate that UCD is widely integrated in the industry.

Based on UCD philosophy and following an overall process defined by ISO standards, a variety of techniques and processes have been developed to facilitate the application and improve the process (see Table 1).

Table 1: UCD methods, techniques and knowledge overview

Methods	Techniques	Supporting artefacts
• Scenario-based design [Carroll 2000] • Goal-directed design [Cooper 1999] • Contextual design [Holtzbatt and Beyer 1998] • Participatory design [Ehn 1988]	• Personas • Demographic analysis • Task analysis • Scenarios • Workflow modeling • Context analysis • Rapid prototyping	• Design guidelines • Principles • Style sheets • Patterns

Similarly, there has been a noticeable effort to provide UCD specialists with software tool support. Most currently available tools target a single aspect of the process. In general, we can divide available tools into two categories: those concentrating on requirements analysis and those concentrating on different stages of design and implementation. Tools like CTTE and Morae (see Table 2) are targeted towards one or a set of activities conducted during demographic and contextual analysis. On the other hand, d.tools and DENIM are targeted towards early prototyping efforts.

Table 2: A non-exhaustive list of software supporting UCD

Tool Name	**Short description**	**Supported technique or method**	**Source**
DENIM and SILK	DENIM is an outgrowth of the original SILK project, a pen-based sketching tool for designing user interfaces. DENIM is a system that helps web site designers in the early stages of design. DENIM supports sketching input, allows design at different refinement levels, and unifies the levels through zooming.	Rapid prototyping	http://dub.washington.edu/denim/research/ OR http://dub.washington.edu/projects/denim/
CTTE	CTTE provides thorough support for developing and analyzing task models of cooperative applications.	Task analysis	http://giove.cnuce.cnr.it/ctte.html
EUTERPE	Euterpe is a Task Analysis tool developed	Task analysis	[Dead or not accessible]

Tool Name	Short description	Supported technique or method	Source
	at the Vrije Universiteit Amsterdam. Euterpe helps to build task trees, object hierarchies and other important concepts such as event and roles. Templates allow detailed information to be specified and multimedia can be attached to concepts to clarify their nature.		
Vischeck	Vischeck is a computer simulation of the entire process of human vision. Vischeck simulates colorblind vision.	Evaluation and testing	http://www.vischeck.com/
SansGui	SansGui is modeling and Simulation Environment for developing and deploying scientific and engineering simulators without writing any Graphical User Interface code.	Development	http://protodesign-inc.com/sansgui.htm
Morae	Morae is a software for usability testing and user experience research that helps you identify site and application design problems and share them with stakeholders.	Persona Workflow Evaluation and testing	http://www.techsmith.com/morae.asp
Observer XT	The Observer is the professional software package for the collection, analysis and presentation of observational data.	Demographic analysis Persona Context analysis	http://www.noldus.com/site/doc200401012
TaskArchitect	TaskArchitect is a Task Analysis Software designed by human factors specialists to make hierarchical task analysis easy.	Task analysis	http://www.taskarchitect.com/
CDTools	CDTools is a suite of modules targeted at easy analysis, maintenance, and sharing of customer field data.	Demographic analysis Context analysis	http://www.incontextdesign.com/cdtools/
d.tools	d.tools is a hardware and software system that enables designers to rapidly prototype the bits (the form) and the atoms (the interaction model) of physical user interfaces	Rapid prototyping	http://hci.stanford.edu/dtools/

Overall, we notice no tools that attempt to integrate all UCD into a single package. Therefore, the designer applying UCD is responsible for integration and conversion of information obtained in different tools and utilities at all stages of the design. This implies that the designer must rely on his/her expertise in order to disseminate the relationships and dependencies between information at different stages of the design. It is also important to mention that a designer is again responsible for managing and updating all the pertinent documentation every time there is a change.

While such issues can be solved by incorporating a set of regulations, formats etc. into standard practices of a given organization, these solutions often rely on a single resource person and are not always properly applied and followed from project to project. In our opinion, a better solution resides in a tool-supported process incorporating all required artefacts. As a solution, we propose a set of specifications and their implementation as a proof of concept support environment based on an adaptation of a novel UCD process integrating the relationship between user experiences and conceptual design.

2.2 Patterns

Initially, patterns were introduced by the architect Christopher Alexander in the 1970s in his book "A Pattern Language" [Alexander et al. 1977]. Since the publication, the idea of identifying patterns and commonly accepted successful solutions to reoccurring problems has been reused in other domains and areas of expertise.

Amongst others, his idea has strongly influenced computer science. One of the most known applications of Alexander's idea published almost 20 years after the original work is a set of patterns developed by software engineers for object-oriented software development [Gamma et al. 1995]. Since then, design patterns concentrating on all aspects of computer science have been used widely in the community.

2.2.1 Patterns in HCI

Interaction design patterns, also called Human-Computer Interaction (HCI) patterns, have been introduced as a medium to capture, disseminate and reuse proven design knowledge, and to facilitate the design of more usable systems. As a result, they are a key artefact for user-centered design, with interesting ramifications for designing across a variety of contexts. Patterns are toolkit independent, presented in a specific format with defined attributes, and applicable at different levels of abstraction. These levels include the user-task model, navigation model or the concrete presentation of the user interface (UI).

They are a great source of interest not necessarily because they provide novel ideas to the software designers' community, but because of the way that they package already-available design knowledge. More precisely, Alexander has defined a pattern as a three part rule (see table below) that expresses a relationship between a certain context a problem and a solution [Alexander et al. 1977].

Table 3: Alexander's pattern structure

Field	Definition
Title	Title capturing the solution offered by the given pattern.
Asterisks	A symbolic representation marking the significance of the pattern. Where: two asterisks represent a "true invariant", one - pattern progressing towards an invariant, but requiring further work, and no asterisks - no invariant has been established.
Picture	An image "... which shows an archetypal example of that pattern".
Introduction	A section describing the context and providing links to higher level patterns.
◊◊◊	Marks the beginning of the problem
Headline	A summary presenting essential elements of the problem.
Body	A section describing the empirical background of the pattern, the evidence of its validity, range of variation of manifestation.
Solution	A section describing the "... field of physical and social relationships

Field	Definition
	which are required to solve the stated problem in the statement context" as a statement in imperative form.
Diagram	Diagram of the solution. Note that Alexander believed that the solution should always allow for diagrammatic representation.
◊◊◊	Marks the end of the main body of the pattern.
Connections	Links to lower level patterns required to complete this patterns

This way of presenting information to designers promotes reusability of best design practices, and avoids reinventing the wheel each time we develop a conceptual design. Moreover, Alexander's format allows for evolution and linking of patterns into logical sequences related by a solution.

However, the early adoptions of the Alexandrian format in HCI has simplified and sometimes even dropped parts of the definition. For example, currently available and commonly accepted libraries [Tidwell 2005, Welie 2003] still do not have a field similar to "Asterisks" in Alexandrian format (see Appendix B: Commonly Used Pattern Formats). It is only following the growth of interest in the HCI community and rapid development of the first pattern collections, some HCI practitioners proposed to connect individual patterns to create pattern languages. Interaction Design Patterns [Welie 2003, Welie et al. 2000], Experiences [Coram and Lee 1998], and the Common Ground [Tidwell 2002, Tidwell 2003] play a major role in the HCI field and wield significant influence. However, some of these authors until today do not strictly follow the concept of "Connections" between patterns. In fact, most of them use an informal textual description that requires careful reading and interpretation in order to find relations with other patterns.

2.2.2 The challenges in pattern use

In the effort to provide a methodology for pattern applications, Thomas Erickson [2000] proposed using pattern languages as a Lingua Franca for the design of interactive systems, with the purpose of disseminating the HCI body of knowledge. This proposition forced the author to return to the idea of "Connections" proposed in Alexandrian format that have been now defined as pattern relationships.

Pattern-Supported Approach (PSA) [Lafrenière and Granlund 1999] links patterns with the purpose of supporting early system definition and conceptual design derivation. In this direction, the set of design patterns that form a language are also linked to the task model. Consequently, during system definition and task analysis, appropriate design patterns are chosen for the design phase.

In the same vein, UPADE Web Language [Li 2001] defines a set of standard associations available between patterns. Moreover, all patterns are organized and structured into categories simplifying search and analysis process. For example, Stack Page pattern from UPADE defines four types (all that are available) of relationships to another eleven patterns: Super-ordinate, sub-ordinate, neighbouring, and competitor. UPADE patterns are part of a larger effort to incorporate patterns into a design process.

In fact, Pattern-Oriented Design (POD) has been developed [Javahery and Seffah 2002] in order to improve on the ideas proposed in PSA by linking patterns together. POD aims to exploit explicitly the relationships between a set of patterns, as well as between the pattern language and the design process. POD consists of 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. As a result, this process produces a conceptual design that can then be extended towards a given application environment. 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).

However, all the previously described methods rely on the fact that the designer has extensive knowledge of all commonly used pattern libraries. Moreover, all these methods assume that the designer is capable of selecting from a large quantity (sometimes more than 60 patterns per library) of patterns only those applicable in the given context. For example, each step in POD design as described by Javahery and Seffah [2002] requires the designer to perform a selection of a pattern. As a result, the selection process becomes a key aspect in the design methodologies and may strongly influence its results. However, a large set of continuously evolving pattern libraries with different pattern formats, organizations, naming conventions and overlapping patterns do not facilitate this process. As HCI evolves, it becomes even more important to provide the designers with the tools that can increase the reproducibility, traceability and reduce complexity of the pattern selection process in order to improve pattern usage and application.

2.3 Personas

Initially, personas in software design have been introduced by Alan Cooper from marketing. In his original work, Cooper [1999] proposes to the concept of persona as a tool in order to center the work around end users and their needs. In fact, personas are constructed as fictitious characters based on composite archetypes and encapsulating "behavioural data" gathered from empirical analysis and ethnographical studies.

Since persona concept has been evolved from a user archetype, it is important to understand it first. An archetype is a generic, sometimes idealized model of a person, object, or concept. In psychology, an archetype is a model of a person, personality, or behaviour. In a similar vein, personas are defined as fictitious characters representing different user types within a targeted demographic group. Personas are assumed to be in or part of some environment based on known user situations that are then translated into a set of scenarios. In general, a persona description includes a name a user's context, goals and major questions that need answers. Cooper proposes to describe personas in a textual format based on the initial investigation data gathered from interviews and ethnographic studies.

2.3.1 Persona History in HCI

Although Cooper [1999] argues that designing for any one external person is better than trying to design vaguely for everyone, some recent studies [Chapman and Milham 2006; Rönkkö 2005] bring scepticism. In fact, various results have been obtained with this technique and one of the major factors sited by Rönkkö [2005] was team organization and politics. However, the concept by itself is not flawed when the application and/or organization around it are inappropriate. We consider that development of better support and methodologies can improve the application of personas in the field.

In an attempt to improve the concept, Pruitt and Grudin [2003] have carried out two large-scale studies at Microsoft. As a result, they have found several problems with persona as defined by Cooper and used in their studies:

1. Persona characters are not always "believable" since they are not based on thorough data analysis,
2. The information encapsulated in personas is not communicated well, and
3. It is not always clear how personas can be used.

In order to solve these issues authors have proposed to extend and modify the originally proposed method and format by:

- *Promoting ongoing evolution of personas and extended studies of the targeted population,*
- *Promoting extensive evaluations of the constructed data groups and personas,*
- *Extending personas format to incorporate source data references, and*
- *Incorporating personas as one of the key tools inside the development process.*

In summary, Pruitt and Grudin have extended the method described by Cooper with the goal to ensure that constructed personas are representative, applicable and are applied in the process. However, they give little details concerning the format and the representation of personas used. In fact, there is little detail concerning the methods applied to incorporate source data and the strength of relationship between personas and constructed groups during "affinity" sessions. Finally, it is unclear how detailed personas are and what interaction behaviours are addressed.

In an effort to improve on the propositions presented above, Courage and Baxter [2005] have further detailed the format of personas (see table below). A more detailed format ensures that the constructed personas will have all defined components present and be easily identifiable. Similarly, it is proposed to even further extend the described format by creating a so-called "foundation" document (see Appendix D: Personas formats). More specifically, the proposed personas format is extended to contain goals, fears, reference materials and typical activities that would motivate and justify scenarios and future decisions. Moreover, this format assumes that the links to the supporting data are explicitly available in the document itself: the authors have used "Comment" feature in Microsoft® Word.

Table 4: Persona Components (adapted from Courage and Baxter [2005])

Persona Component	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.
Skill Set or 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	Information about user associates, since this could give insight on other stakeholders.
Disabilities	Information about specific physical or mental requirements, such as limited color recognition.
Requirements	User needs and quotes demonstrating what really drives these needs.
Expectations	Information about how the user perceives the system works, and how the user organizes information related to his/her task, domain or job.
Photograph	Include a photograph which fits with the name.

Overall, a large variety of formats have been employed in industry and academia with more or less important details and variations to the components presented above (see Appendix D: Personas formats). However, most of them use only narrative language to describe persona and often do not clearly define the requirements for each component.

2.3.2 The challenges of persona use in design

[Chapman and Milham 2006] have quoted personas to be a "popular method" to model users. Although popular, this is a relatively new and unknown technique to a wide variety of specialists working in HCI or related domains. Moreover, as any new creation it undergoes a rapid cycle of evolution that also influences its use and effectiveness in a design process. Lack of a common standard designed for evolution, unclear usage guidelines and lack of integration into a design process are some of the major challenges encountered while working with persona as a technique.

One of the most commonly known problems with personas resides in the fact that their use in a given project is not always clear to the team members. For example, Blomquist and Arvola [2002] report that less than half of the members knew the personas' names used in the project and most did not recognize their faces. Because most of the members were not aware of whom the personas were, the discussions related to needs (during design and development) were extremely difficult. In fact, scenarios have been reported to play a greater role than personas themselves.

Recent studies [Rönkkö, 2005] have demonstrated that application of personas may be ineffective in rapidly developing product categories. Rönkkö presented an example of a telecommunication market project where the personas have been used during high-level design. However, in the progress many tradeoffs and modifications have been made. In fact, initial requirements have been partially overwritten with modification dictated by the market and competition: "It [Telecommunication Industry] produces artifacts with potential usability, and not the other way around. Designing new, hot and advanced technical components for the mobile device has a higher priority than satisfying pre-identified user groups."

Another commonly cited problem with the application of personas is lack of a clear standard and description of information. In fact, Pruitt and Grudin [2002] and Ndiwalana et al. [2005] report that it is difficult to apply the recommended approaches without a clear understanding of persona identification and characteristic inclusion. In the same vein, few published efforts describe projects using personas with disabilities. While 6.9% of the US population over the age of five has a disability (accounting for almost 19 million of people) and 40.5 % of US population over 65 years old have at least one type of the sensory, physical or mental disability [US Census Bureau 2005], very few projects have attempted to construct and integrate a persona with disabilities. The Treasure Board of Canada with cooperation of industry has provided a document Accessibility Domain Architecture amongst other things containing a Disability/Persona Matrix including 10 personas with unique disabilities used to enhance the accessibility to the Government of Canada's Information Management/Information Technology systems [Accessibility Domain Architecture 2003]. Similarly, Microsoft included accessibility information in their personas, but did not create a "real" disabled persona [Pruitt and Grudin 2003 (b)].

Overall, the persona technique becomes more and more popular and the need for a common clearly defined standard becomes more and more important. There is a clear need in all types of organizations for a format that will take into account all realities of life and will allow for future evolution.

3. Process Modelling

We have explored two of the largely "popular" techniques in the HCI community: Personas and patterns. Both are relatively new and lack a common and clear standard. Some of the usage and application problems recently discovered with these two techniques suggest that a set of improvements should be made in attempt to increase their effectiveness.

In a search of a solution, we have established a set of requirements that need to be satisfied. At the same time, we have analyzed results of a proof-of-concept study proposing a novel framework incorporating both persona and patterns. Based on our findings, we have conducted extensive research in order to propose concrete solutions to previously established requirements. Then, we have constructed a process and a supporting tool. Finally we have conducted a validation study with a goal to uncover limitations of the proposed solutions.

3.1 Required framework

In the previous sections, we have briefly discussed patterns and personas. In general, we have found that both tools lack a common standard and have not reached their maturity yet. Moreover, some researchers [Chapman and Milham, 2006; Rönkkö, 2005] have found that the structure and policies within organizations can largely affect the success of these tools. We propose to solve these issues by refining available formats and tightly integrating them into a design process.

In fact, we believe that a design process should be developed in order to ensure that both personas and patterns are used as primary tools and are required for progress. Moreover, it is also suitable to reduce the load on the designer while working with patterns by modifying their format and/or automating the selection process. There is a need for a process, as well as a persona and a pattern format that allows for automation in order to support specialists during their design activities.

3.1.1 Process and artefacts

Let us now consider, jointly both personas and pattern formats. One common trait unites both tools; both of them are based on a narrative description. In fact, all surveyed pattern and persona formats are relying on the ability of the specialist to clearly describe and the ability of the designers to properly interpret large paragraphs of text. For example, one of the most popular pattern libraries [Tidwell 2005] uses from 2 to 3 pages of the printed text to describe a single pattern. Similarly, commonly used persona descriptions use one to two pages of textual descriptions that a designer needs to analyze, understand and assimilate. As a result, before these tools can be integrated into design as key elements, a substantial effort should be made to improve the formats by reducing the interpretation error.

We consider that both personas and pattern formats should have a set of discrete variables that equivalently describe them. In our understanding, a discrete variable is a facet of a pattern that has a clearly defined scope and accepts a small range (typically lower than 10) of values. This representation will aid in two aspects: reducing interpretation errors and allow for future automation or software tool support. When we compare unformatted narrative and the same text separated in paragraphs each with clear title, the second is much easier to analyze and understand. In general, any format will be prone to interpretation from the moment when it uses narrative text in some part of it, like in variable definitions. However, a rigid structure will ensure that a reader has the same format across all the descriptions and the same information is located at the same place. This facilitates interpretation, comparison and analysis of available information. Similarly, any object, part of a larger process, described as a set of discrete variables, can be incorporated in a computer environment supporting a part or the entire process. However, we anticipate that there is still a strong need for textual descriptions. We consider that narrative descriptions should be used in order to explain the values selected for discrete variables. As a result, *both personas and pattern formats should be augmented with a section containing discrete variables presenting an alternative view on information contained in raw text descriptions*.

Now let us consider the requirements related to the process and the two discussed tools. As it has been noted earlier, personas have sometimes failed because they have not been used as a core component of the design process. As mentioned above, we have discussed few cases when personas have been improperly used or have not been applied through all the stages of the design process. Therefore, there is a clear need for a process that will ensure that the information encapsulated in personas is used in all phases of the design and will evolve, as the content will be re-examined given new information. Consequently, in order to increase the effectiveness of personas, *a process must be* *designed to ensure that without proper understanding and use of the personas the progress is impossible; thus, making this tool a key artefact in the design.*

3.1.2 Support Environment

We have previously described both patterns and personas and have noted a lack of standards. However, another major problem has not been mentioned. Any UCD method that applies personas and patterns creates a heavy informational load on the designers. In order to construct a persona, a designer usually conducts a set of studies with a relatively large group of users; then this information must be analyzed and interpreted. Similarly, at a design phase, a designer using patterns will have to select from multiple frequently updated libraries containing up to 100 patterns each. As a result, concrete design solutions are dependant on the designers experience in the analysis of large sets of data and most importantly perseverance in search of or knowledge of patterns and libraries. In order to reduce the involvement of the designer's personal characteristics, *a supporting environment is needed that will facilitate manipulation of large sets of data and reduce the scope of analysis based on a project under consideration.*

3.2 Testing initial hypothesis

Initially, a set of experts have developed a prototype of the framework based on most commonly used and accepted standards in the field (see Figure 2). In order to evaluate the framework, a proof-of-concept study has been conducted with a bioinformatics information portal – NCBI (National Center for Biotechnology Information) website [NCBI 2005], a well-established bioinformatics portal including specialized tools and databases. It is a popular bioinformatics resource for a variety of specialists in all fields related with natural sciences.

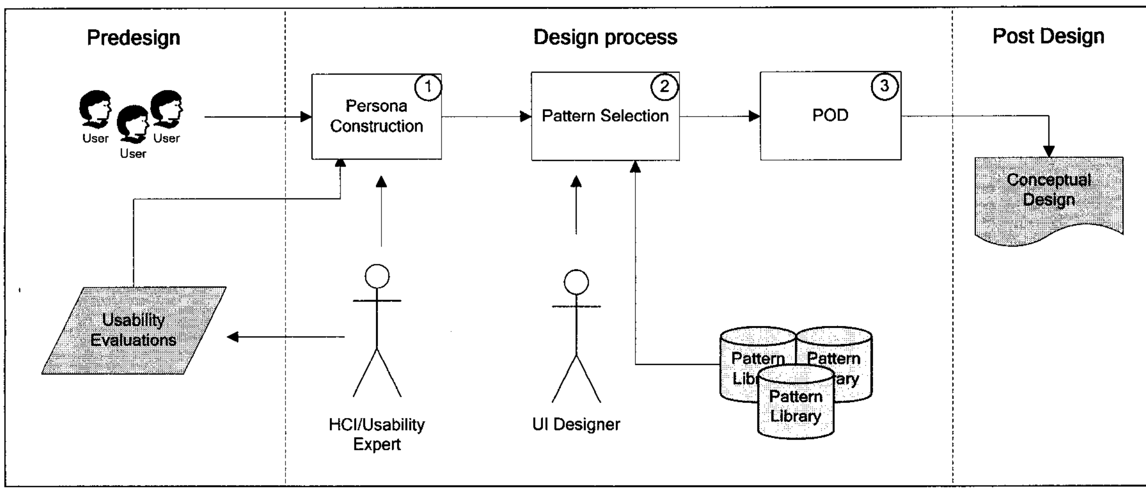


Figure 2: Preliminary Persona to Pattern Framework

The usability study consisted of two phases with 16 and 23 users involved in predesign and post design usability testing respectively. The predesign phase consisted of persona modeling with help of psychometric and heuristic techniques. The results have them been used to select a set of patterns and construct a prototype of the new design using POD techniques. Finally, in the post design phase, the following evaluation techniques have been used to perform a comparative study between the two designs: task based evaluation, structured questionnaires, think-aloud protocol and open-ended interviews.

In essence, the goals were:

1. "*Evaluate whether the application of the framework results in more usable systems,*" - as usability is a broad topic and many definitions exist we have attempted to reach this goal by using the definition from ISO standard [Abran et al.2003].

2. "*Discover the process used by designers when going from personas to patterns,*"- process discovery was attempted through various information gathering activities.
3. "*Understand the limitations of the framework.*" - is achieved through analysis of information collected when attempting to satisfy the other goals.

3.3 Knowledge acquisition

Following results obtained from the NCBI preliminary study, we have used a set of elicitation techniques in attempt to gather additional knowledge and fill-in the gaps in the experimental framework. Our main goal was to elicit “what, how and why”. That is, we have attempted to gather experts’ knowledge on the domain and their strategies for problem solving including detailed rationale. We have concentrated on two major problems: 1) input format and content and 2) process of linking user experiences with patterns. The first goal consists of establishing the format, the content (required and optional) and rules for future modification of patterns, personas and users. The second goal consists of exploring and defining a strategy or a process that would allow a designer to relate between user experience and patterns.

In order to achieve our goals, we have used three elicitation techniques (see Table 5). These techniques were selected based on feasibility, available time and resources.

Table 5: Used knowledge elicitation techniques

Technique	Description
Document analysis	Examination of guidelines, process description and other relevant documentation on processes involving manipulation of personas and patterns. Main focus is on discovering process steps and artifacts used.

Technique	Description
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.

We have alternated the techniques with a goal to obtain best results possible. For example, semi-structured interviews have been used in the beginning of the knowledge gathering process in order to define the scope and main directions of the research. At the same time, they have been used at the end in order to validate the combined knowledge from all previous resources.

Table 6: Knowledge acquisition time line

Technique	Time Line
Semi-Structured Interviews	
Document Analysis	
Expert focus group	

Overall, information-gathering techniques have been structured with a goal to validate not only our raw findings but also our interpretation and the combined results.

3.4 Developing supporting environment

At this stage of research, we have gathered all previously acquired knowledge in order to construct the framework. We have divided our problem is a set of sub-steps:

1. Develop an improved persona and pattern formats.
2. Analyze and detail process of pattern selection based on user experiences.
3. Construct a process linking user experiences and patterns.
4. Construct a supporting environment automating most of the process.

During each sub-step we analyzed previously gathered information in order to synthesize relevant knowledge. By building on the state of the art processes and formats we have attempted to construct a lightweight and pragmatic framework with possibility for future automation.

This included various modifications to both persona and patterns allowing for computer analysis. We have performed two iterations. Insight from a formal focus group with five experts has been used to construct initial models, that then where analyzed and refined during an interview with a cognitive psychologist.

At the same time, a detailed process description in form of discrete sequential steps has been constructed. Based on the analysis of current practice [Cooper and Reimann 2003; Courage and Baxter 2005] and interviews with two experts (UI designer and cognitive psychologist), and insight gained during NCBI study we have described a design process typically carried out by experts intuitively.

Finally, we have analyzed available process information with a goal to develop a supporting environment automating most tedious steps. Some of the design steps have been closely analyzed and possible automation solutions proposed in the literature and other fields of study have been examined [Suryavanshi et al. 2005]. However, the tool was constructed with the idea of creativity in mind giving the designer a choice of performing operations manually.

3.5 Testing and validating the findings

In order to evaluate the proposed framework we have conducted a study consisting of a design of a functional prototype of protein visualisation tool. We have applied P2P Process, supported by P2PMapper tool with a goal to construct a conceptual design. This design then has been implemented to construct a fully functional prototype [Javahery et al. 2007(c)]. Finally, we performed an evaluation of the new design. Similar to the evaluation described in Section 4.3, we have conducted a comparative randomized study using task-based evaluation and open-ended interviews.

Given the fact that we where attempting to create a new user interface design rather then redesign an existing interface, we have carried out all predesign empirical studies on tools similar to our target design. Usability inquiries where conducted in form of field studies and user observations with questionnaire administration on two Bioinformatics visualization tools, Cn3D [2005] and AND-Viewer [Gros et al. 2005].

The P2P Process has been used in order to construct persona and develop a conceptual design. In fact, findings from usability inquiries have been entered in P2P Tool in order to construct persona and select a set of relevant design patterns. Based on the set of patterns a conceptual design and a functional prototype have been constructed.

Then, we performed an evaluation of the new design. Similar to the evaluation described in Section 4.3, we have conducted a comparative randomized study using task-based evaluation and open-ended interviews.

Finally, we have performed informal interviews with experts in order to evaluate the limitations of the P2P Tool used in the scope of P2P Process.

4. NCBI study: testing initial hypothesis

This chapter discusses the results of the study that has given us the fuel and a basis for the research. The results obtained from the study have been used to narrow the scope of the research and ensure that the framework used in this study is sufficiently promising to continue the research.

This study has started as part of the research on *Pattern Oriented Design for Interactive Systems* [Javahery 2003]. In the scope of the current (our) research, we have analyzed and compiled the results with a perspective towards eliciting a process for persona centered pattern oriented design. Moreover, we have completed post design evaluations to complement already available data. Therefore, in this section we will present a concise description of the study necessary for the future understanding of our research. For a more detailed analysis please see [Javahery 2007 (a)(b)].

4.1 Predesign process

The main goal of the predesign process was to construct a set of representative personas encapsulating most of the essential knowledge about real users. We have started by performing domain analysis and a set of ethnographic interviews. The results obtained have allowed for construction of an initial set of personas; three personas have been constructed (see Appendix A: Persona Set (NCBI study)). These personas were constructed based on expert analysis of a set of attributes that seemed to be most influential: age, work environment and application experience of real users.

Once the initial set of personas was constructed, a set of usability studies was performed with a goal to obtain additional knowledge on behaviour and performance of the users in order to increase the effectiveness of our personas [Pruitt and Grudin 2003]. More precisely, psychometric assessments and heuristic evaluation have been used.

Psychometric assessment consisted of a questionnaire with 31 items [Javahery 2007] that has been distributed to 16 NCBI users. At the same time, user interaction behaviour and needs have been assessed. The main tasks users were performing varied extensively and seemed to be dependent on the application experience. Based on these variations and after expert analysis two major groups started emerging: Novice and Expert users. In fact, Table 7 below depicts the differences between these groups.

Table 7: Interaction Behaviour and Needs of NCBI Users by Groups

Group	Needs	Interaction behaviour
Novice	▪ Guidance ▪ Simple Navigation	▪ Literature searching ▪ Information gathering and use of tools, such as BLAST
Expert	▪ Control ▪ Task Efficiency	▪ Follow a scientific process whereby they were repeat users of specific tools.

These results have suggested that there maybe only two major groups of users. A deeper analysis of the newly gathered information showed that Persona 3, Dr. Thomas Johnson, had a majority of needs and behaviours already represented by the other two personas. Since the goal is to have the smallest possible representative set of personas, Persona 3 was eliminated. This resulted in two final personas that have been called: *The Novice User* and *The Expert User* (see Appendix A: Persona Set (NCBI study)). Although, their names have been selected in order to suggest their experience, it is important to remember that this set is a result of the large analysis conducted by experts and involving a large set of criteria (including but not limited to user experience).

4.2 Design process

Based on the final set of personas and the information obtained from heuristic evaluations usability experts have selected a set of patterns to be used in the redesign of the website. We must note that the patterns have been selected based on the best knowledge of the expert and his/her interpretation of the available information. That is, none of the experts have been able to clearly specify the process that has been used to obtain a list of patterns. However, they have been able to justify their choices by linking persona and pattern descriptions.

After pattern selection, POD approach has been applied in order to combine the patterns and construct a conceptual design. Given the size, the complexity and the dynamics of the website, separate designs have been rejected on a basis of technical and maintenance specific difficulties. Therefore, the experts have settled for a compromise searching for a design usable by both novice and expert [Javahery et al. 2007(a)(b)].

For example, Figure 3 and Figure 4 demonstrate a conceptual design of a home page and an implementation of this design respectively. Recall that POD does not result in implementation design but rather in a conceptual design as it is presented below. A conceptual design can still be largely influenced by color scheme, character styles and other visual attributes during the implementation phase.

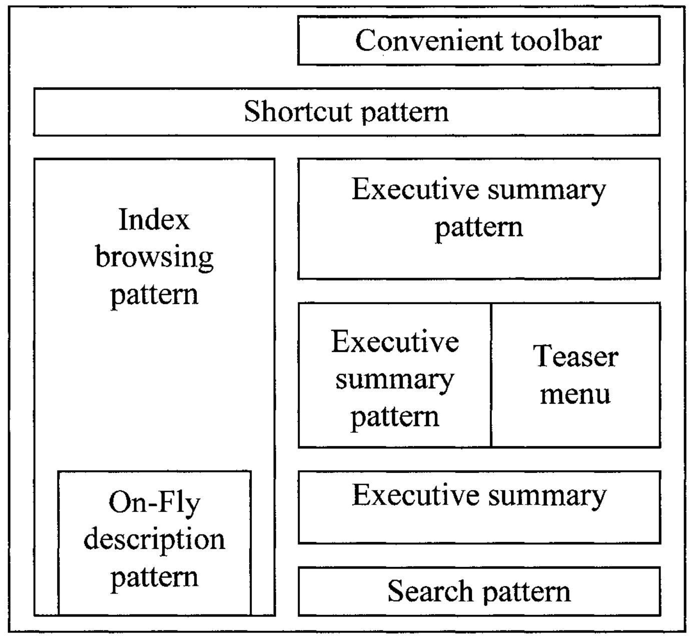


Figure 3: Conceptual design of a Home Page of NCBI Website

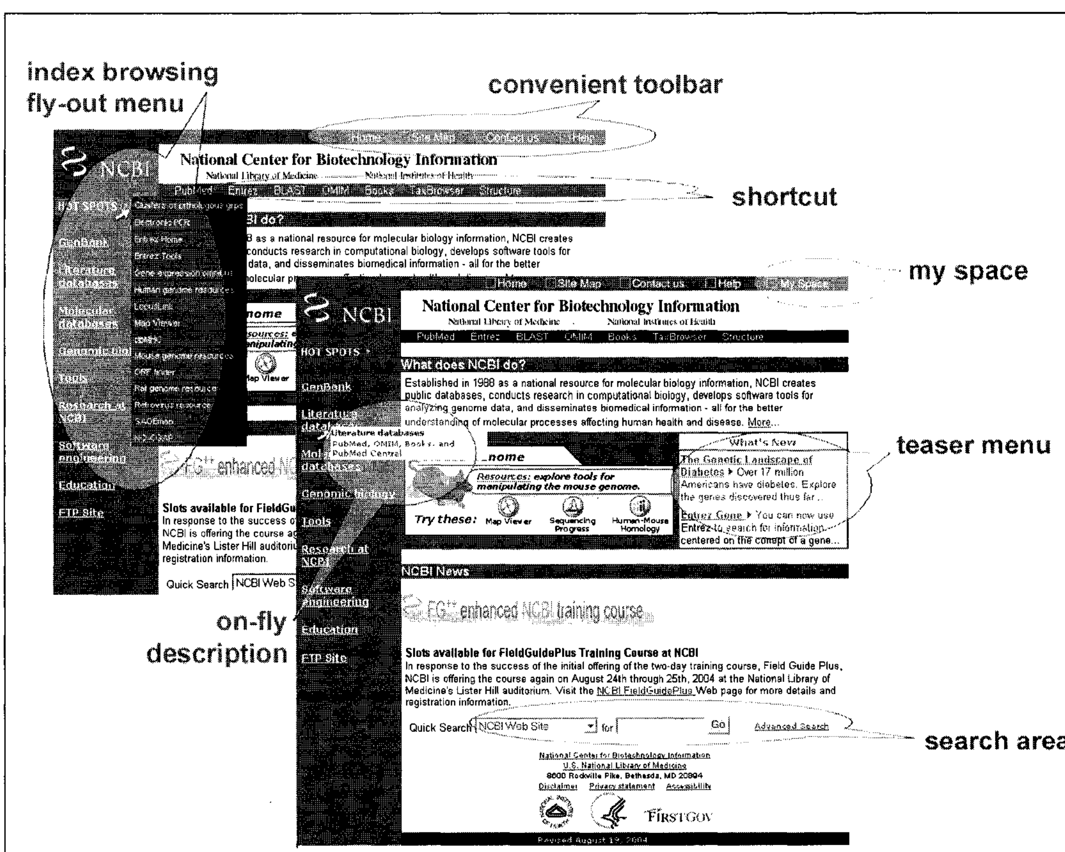


Figure 4: Implementation of a Conceptual Design of a Home Page of NCBI Website

However, by following general guidelines for a successful design, a set of designers is expected to achieve a similar level of usability. In the case of NCBI, the experts have chosen to apply a similar color scheme to the one available at this time on the original NCBI website in order to facilitate the transition for the frequent users.

4.3 Post design evaluation and results

In this phase, we have conducted a comparative study with the current NCBI site and newly constructed design. We had 23 participants, where 19 would fit into *Novice User* and 4 into *Expert User* group, as defined in the final set of personas. The users participating in this evaluation were selected based on an informal interview ensuring that they fit the study. Similar to the population, the selected sample came from a variety of educational and professional backgrounds.

Overall, we have attempted to select a sample that corresponded to the original population. However, a large misbalance can be seen in the Expert User group. This is due to a small size of the group and has been taken into account during the evaluation. More precisely, users belonging to this group have participated in qualitative and not quantitative evaluation. That is, for the Expert User group we have used the coaching method while exploring the software with structured and open-ended interviews. They were given time to explore both versions of the website and were asked a set of questions relating to their experience.

On the other hand, for the Novice User group we have used a between-subjects design, where each participant was assigned to a different condition. That is, we have selected one variable with two possible values and randomly separated users into two groups, one for each value [Dix et al 2003]. In our case, the variable was the design type, with two possible values: original or "old" design and the newly constructed design or "new" design. Since each user was testing only one type of the design without any prior knowledge about it, we have been able to control the learning effect.

Novice users were given four common and basic tasks to perform on the website. Their interactions have been recorded and marked by using Morae Usability Testing Tool [Morae 2006]. After the users completed the testing, we have performed individual open-ended interviews in order to elicit the subjective impression, appreciations and other relevant details about the user's experience.

Once all the evaluations have been completed, we have assessed our quantitative hypotheses by performing a series of statistical tests and comparing average group values. More precisely, we have selected ANOVA [1] to assess if the mean values obtained are significantly statistically different. In addition to statistical significance, we have computed effect size, eta-squared (η^2), which is a measure of the magnitude of the effect of a difference that is independent of sample size[2]. In HCI, due to commonly small sizes of studies, effect size has been found to be often more appropriate [Landauer 1997]. In

---

[1] ANOVA - Analysis of variance is a collection of statistical models and procedures analyzing significance of differences between means.

[2] Eta-squared - A calculation in the t-test family that is analogous to r^2 in correlation and regression. This number can be interpreted in exactly the same way as r^2. Therefore an $\eta^2 = .367$ means that 36.7% of the variability in the dependent variable can be explained or accounted for by, the independent variable. Calculated by $\eta^2 = \frac{t^2}{t^2 + df}$

general the greater the value of η^2 the greater is the effect. Some HCI practitioners use the following metrics for interpreting η^2: 0.01 is a small effect, 0.06 is medium, and 0.14 is large [McGrenere 2002]. In our analysis, we have refrained from using a particular metric for η^2. Instead, we compared the suggestions from ANOVA and size effect in order to arrive to a conclusion.

Finally, if the test has confirmed our expectations, we compared the means and the variations in order to assess an overall difference between the two values.

4.3.1 Time Improvement

As one of our hypotheses, we have stated that we expect to have a significant improvement in time required to complete a task between an original design and the one we have produced using POD. In order to assess our hypothesis we have performed five ANOVA tests:

- Tests 1 to 4 were performed by comparing the time required to complete a task in the case of the new and old designs. Each test compared values within a given task.
- Test 5 was performed by comparing the time required to complete all four tasks in the case of the new and old designs.

Table 8: F values of ANOVA single factor tests for task times of new vs. old design.

Task #	F	p	eta-squared	Average time difference (in %)*
1**	4.75	0.05	0.25	-54.80
2	8.47	0.01	0.35	-69.72
3	8.77	0.01	0.35	-67.26

4	7.43	0.01	0.32	-41.67
all	27.92	0.00	0.67	-55.61

* Average time difference (in %) was calculated by assessing: $\frac{mean_{new} - mean_{old}}{mean_{old}} * 100$

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

For all tasks, the results ($p \leq 0.05$) suggest that our hypothesis is true. Moreover, we can see the same results when we compare the total times of both designs. 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, we can see that on average there was an improvement of more than 40% in times when considering the average time required to complete the task. At the same time, if we consider that all four tasks represent a relevant sequence of actions in a common situation, we note that overall improvement was of more than 55%. Therefore, the results suggest that there was a significant improvement in the time required to complete a task in the new design.

4.3.2 Success/Failure Rates

Although we have predicted that there should be a significant improvement on the failure rates, ANOVA test has failed to show any relation between the type of design and the failure rates. As we can see in the table below the tests for task 1 and 3 show that there is no statistically significant effect of proposed framework on the failure rates. Although eta-square indicates a relative low effect, we cannot draw a conclusion from this data. However, when we considered failure rates for all the tasks together, we find that

ANOVA and eta-squared show that there is a relatively strong, positive and statistically significant effect of POD on the failure rates: F (1, 16) = 6.4, $p<0.05$, $\eta^2 = 0.29$.

Table 9: F values of ANOVA single factor tests for failure rates of new vs. old design

Task #	F	p	eta-squared
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 have been detected for the task 2 and 4 in any type of the design.

Therefore, we cannot affirm that the POD design offered a significant improvement in failure rates. Moreover, analysis suggests that further investigation by increasing the number of participants may provide additional insight and confirm or reject our hypothesis.

4.3.3 Novice User Satisfaction

The satisfaction ratings have been based on a five point scale where 1 = totally dissatisfied and 5 = completely satisfied with the design. The results were compiled based on the individual comments of the users that were asked to rate their satisfaction after performing four requested tasks. Moreover, the comments on possible improvements were also considered as part of the satisfaction measure.

The results of study (F(1, 16)=11.53, $p<0.05$, $\eta^2 = 0.42$), suggest that there is a significant difference in the satisfaction ratings due to the variation of the design type (new vs. old). Moreover, when we consider an average ratings of both designs (new=3.44, variance=1.28; old=1.89, variance=0.61) we find that overall the users were almost two times more satisfied with the new design than with old one.

It is also interesting to note that the variance in the case of old design is of 0.61, which indicates that most of the users had a similar opinion: they did not like it. On the other hand, the new design has spawned much more variation (variance=1.28) in opinions; 45% users were satisfied with the design (rating of 4 and 5), 22% were dissatisfied (rating of 2) and 33% were undecided (rating of 3). These results suggest that some users have noticed a relatively serious issue in the new design. Therefore, although we have achieved a higher satisfaction rate with the new design, the variance of the opinions indicates that more analysis should be performed in order to search for remaining issues triggering dissatisfaction.

4.3.4 Expert User Post-Interview Results

Expert users have been using the website for a large period and have become accustomed to its current design. Consequently, some of them preferred the "old" design for their daily tasks while still mentioning that the "new" design is "easier on eyes." However, we must note that the sample size is too small to draw any conclusions. In fact, a much larger sample size is required in order to establish tendencies and maybe predict possible interaction of expert users with the "new" design. Given that expert users have been tinted by transfer of knowledge and learning effect, a study over a sufficiently large period is essential in order to establish how fast and how well these users will adopt to a new design.

4.4 Observations

We have performed a set of informal and semi-structured interviews with specialists in order to extract additional details concerning the framework applied during the NCBI study. During interviews, our goal was to draw in point form a process that would allow an outside expert to trace back each decision and eventually be able to follow the process and come up with a similar design.

However, we have found that the most experts apply a highly nuanced intuitive process that they are not always able to describe. More precisely, we have been specifically interested in pattern selection process. Pattern selection process has been identified as the least covered and described step in the entire design cycle in HCI. Our interviews have shown that:

1. In order to reduce the memory load experts select one or two pattern libraries. Some times, they prefer selecting categories (2 or 3) of patterns from particular libraries.
2. The experts heavily rely on their previous experience when selecting a particular library of patterns to work with.
3. The process of pattern selection based on persona is fuzzy and relies on expert impressions, interpretations and previous experiences.

Therefore, it is possible that different experts come up with different conceptual designs. Moreover, it is hard or even impossible to trace back the decisions that have lead to a particular pattern selection.

While searching for pattern selection process, we have discovered that personas, as they have been used in the study and in the current UCD practices, lead to some differences in interpretation. Experts tended to see personas differently. More precisely, experts as human beings have been reading, interpreting and applying textual descriptions within personas in slightly variable ways. In fact, personas are in essence a text that generates some imaginative image of the user and his/her behaviour. However, the imagination is strongly influenced by personal experiences and cultural background. Therefore, when time comes to find a compromise (in pattern selection for example); there is a possibility for an error based on misinterpretation of the source information.

4.5 Summary of the NCBI study

In previous sections, we have described a case study where an experimental framework narrowing the "gap" between the user experiences and conceptual design was used. The proposed framework was centered on personas and patterns with a strong emphasis on empirical studies in the predesign phase. It was applied in a scope of a redesign of a part of a large Bioinformatics website. We have attempted to evaluate the applicability of the framework, whether the framework application resulted in a more usable system and the limitations of the proposed framework.

We found that the framework encapsulated most commonly accepted design practices and UCD principles organized in some logical sequence and structure. Usage of persona as one of the main artefacts has reduced the weight of the process. Similarly, patterns, as pieces of the best practice knowledge, when used as building blocks during conceptual design, potentially accelerated the design by favouring reuse. As a result, the proposed framework has facilitated the process and the design overall.

Additionally, we have performed a comparative usability study with our prototype and original framework in order to verify our assumptions. We have found a statistically significant improvement in task duration times and user satisfaction. We have noted an improvement of more than 55% in task duration times and the users were almost twice more satisfied with our prototype. However, expert users demonstrated a divided experience that led us to conclude that further investigation is required.

Moreover, we have extracted additional information related to framework application by specialists. We found that the links established by specialists between user experiences and design solutions (in form of patterns) were based on qualitative data, assessed using "best" experiences within a specific context of use. In essence, the experts have applied an intuitive reasoning method incorporating multifaceted knowledge collected in the current study and gathered from their previous experiences. At the same time we have found that personas, as textual descriptions lack precision and may be interpreted differently by different experts. More precisely, currently available format for persona specification is a narrative text that relies greatly on common cultural and social believes and narrative skills of the reader and writer. In order to improve the proposed framework, we need to refine the persona description and brake down the pattern selection process into a set of discrete step that will not rely on experts experience as much as it is possible.

Overall, the experience conducted with the preliminary framework has been successful. Moreover, it has provided the leads towards future refinement and development. Therefore, we have concluded that a deeper analysis and research may result in a process that will allow effective application of persona and patterns as main artefacts in the design process.

5. Persona and Pattern Models

In previous sections, we have considered global requirements for our framework and analyzed a preliminary model during NCBI study. Based on our findings, we propose a set of solutions that in our opinion constitute the basis for a traceable and systematic design process. In this chapter, we propose a set of additions to persona and pattern format. Moreover, we analyze a process of pattern selection based on persona.

5.1 User and Persona

Following our earlier expectations, we have found that persona and user description formats should be the same. Persona is a representation of a set of users and should follow the same format. Therefore, we have concentrated on exploring user/persona and pattern formats. First, we have conducted a document analysis that provided us with initial set of variables. Then these variables were refined through semi-structured interviews and focus groups.

In general, personas are presented in a narrative form, however in this format it is hard to establish content requirements and effectively use these descriptions in computerized supporting environments. We find that computer supported processes are more accessible and adoptable by a wider population. A newly proposed model must be defined in a format that will allow manipulation by computers. Therefore, we have attempted to gather all information that would allow us to construct human and computer readable model.

The variables were selected based on a set of studies conducted during research. More precisely, the initial set was constructed from results of a set of literature reviews: all variables have been included. Then the initial set has been refined based on the close interaction between three experts in the field. The main goal of the refinement was to remove duplicate variables and consider if some variables can be split or combined. At the same time, entirely new suggestions were considered with the supporting proof from literature and common practice. Since the refinement process is naturally an endless spiral, we have defined the stop point as the moment when there are no more entirely new suggestions to add or there is no substantial proof for any of the new suggestion proposed.

The scales, on the other hand, were selected based on the most commonly accepted format used by the experts in the field where the variable comes from. For example, Cognitive Style variable comes from psychology field and many formats are used to describe it. Most common one is the Myers-Briggs Type Indicator [Jung 1971] that has four subparts: Introvert vs. Extrovert, Judging vs. Perceiving, Thinking vs. Feeling and Sensing vs. Intuition. This resulted in a set of attributes grouped into five different categories: (1) Demographics, (2) Knowledge and Experience, (3) Psychological Profile and Needs, (4) Attitude and Motivation, and (5) Special Needs. Each variable is defined on a binary, 5-point or 7-point scale. For example, variables like gender are binary with possible values "male" or "female". On the other hand, variables such as domain experience are defined on 5-point semantic-differential scale: 0=none, 1=basic 2=average, 3=advanced and 4=expert. Similarly, some variables, such as income, have been defined on 7-point scale following commonly accepted standards in demographics.

Finally, two variables in the proposed set (Disabilities and Special Groups) can contain an array of values.

Table 10: User Variables

User Variable	**Description**	**Range of Values**
1. Demographics		
Age	Age group or range	{toddlers, children, adolescents, young adults, mature adults, seniors, elderly}[3]
Gender	Gender	{male, female}
Income level	Family income, where low income defined as < 50% of median[4]	low ↔ high
2. Knowledge and Experience		
Computer experience	Where basic experience is working knowledge of office systems	none ↔ expert
Domain Experience	Experience in technical or business function supported by product	none ↔ expert
Education level	Formal training and education	{none, elementary, high school, vocational, college, undergrad, advanced }
Linguistic ability	Knowledge of product language	none↔ fluent
Literacy	Ability to read, write, use numbers, and handle obtained information	illiterate ↔ fully-functional literacy
Product Experience	Experience with product or with similar software products	none ↔ expert
3. Psychological Profile and Needs		
Behaviour to features	Behaviour and interaction style towards software features	feature shy ↔ feature keen
Control	Amount of control user needs when interacting with the product[5]	low ↔ high
Guidance	Amount of guidance required when interacting with the product	low ↔ high
Initiative taking	Initiative taking habits of user when interacting with the product[5]	reactive ↔ proactive
Learning speed	Rate this user's learning abilities in general (slow learner/fast learner)	low ↔ high
Learning style	Primary learning style of user	{auditory, visual, kinesthetic}
Learning support	Learning support required when interacting with the product[5]	low ↔ high
Validation of decisions	Validation of decisions required when interacting with the product[5]	low ↔ high

[3] Toddlers (0-4), children (5-14), adolescents (15-19), young adults (20-34), mature adults (35-39), seniors (60-74), elderly people (75 and over).

[4] Middle point on the 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$)

[5] or with similar products.

User Variable	Description	Range of Values
4. Attitude and Motivation		
IT attitude	Attitude to information technology in general	negative ↔ positive
Motivation level	Motivation to use the system	low ↔ high
5. Special Needs		
Disabilities[6]	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}

Overall, the proposed set maybe extended to include the latest findings in the HCI or related fields. However, in all cases new variable descriptions must at least follow the format proposed above.

That is, all variables should have a name, a description and a limited range of values. Note that proposed set of variables suggests that value range should contain between 5 and 7 values. Moreover, we found that grouping variables into categories helps ensuring that all aspects have been covered in the proposed format. Thus, every newly added variable must belong to a group. It is also essential to identify why the information provided by this variable is needed and how it can be used. While a given variable may overlap with existing ones, a designer must ensure that it describes some essential new aspect that cannot be deduced by applying basic logic from already available variables. For example, a user with low computer and domain experience generally requires high levels of guidance. However, high level of guidance may also be needed for users with

[6]Users may belong to more than one group at the same time. In contrast with other variables, "disabilities" is an array of values.

high experience because of other personal factors. This variable may overlap with others but still describes a unique facet of a user that cannot be easily deduced.

In summary, we must note that the set provided above is not complete and may not be even sufficient for a large variety of projects. However, it is a first step towards a clearly defined discrete format that strives away from simple raw textual descriptions. We have performed a large analysis of available information from a variety of sources including literature [Kirakowski and Corbett 1993; Aaker 2004; Statscanada 2006] and hands-on expertise accumulated by a variety of specialists in the field. This resulted in a set of variables that describe commonly identified facets of a user or personas. The resulting format satisfies our objectives by providing an alternative description of users and personas that can be later incorporated in and analyzed by software tools.

5.2 Pattern Model

In the process used during NCBI study, patterns are used as an input during selection phase. However, we must note that a small set of them will also be used in the POD phase. Moreover, a resulting conceptual design is a composition of patterns. When constructing and refining pattern description, we must consider how this artefact will be used in the process.

In current practice, patterns are represented following Alexandrian format of "context, problem, and solution" [Alexander et al., 1977]. Some libraries also include forces, examples, related patterns, short descriptions etc. [Tidwell, 2005, Welie, 2003]. Moreover, some proprietary and commercially developed formats brake down pattern description into even smaller blocks of information [Vetere et al. 2006]. However, all these formats are constructed with the goal to provide the designer with necessary tools required to assess the applicability of a pattern for a particular design. Although often misinterpreted, common language is used to deliver all this information. Therefore, designers rely on their previous experiences to infer required information based on their interpretation of provided content. For example, designers need to infer design criteria (such as structure, navigation etc.) addressed by a given pattern based on the information presented in forces, examples and other relevant sections of pattern description.

Following our analysis, we have found that a pattern should contain a *short description* attribute. In fact, such a field has been used in Tidwell book [Tidwell 2005] in Table of contents. Although Tidwell does not always follow this rule, short description must always contain keywords from context problem and solution organized in a short sentence or a small set of sentences. Similar to table of contents, it can be used to facilitate pattern selection process by providing short and concise descriptions on fly when and where required for immediate reference. Moreover, following UPADE library [2004] and pattern language requirements described by Todd et al. [2004], we consider that a pattern should contain *relationships* attribute. This attribute should include all relationships existing for a given pattern. This attribute is particularly useful when applying POD method for pattern composition.

While it is essential to describe further the *relationships* attribute, we must note that this attribute may contain variable information depending on the UCD method applied. In general, this attribute should facilitate navigation from pattern to pattern and at least allow logical inclusion or exclusion. It must contain information on the type of relationship and the pattern that it relates to. A large variety of relationship types may be created. However, a designer must strive for a smallest satisfactory number possible. A small amount of relationship types will reduce learning curve when a novice starts using the format. At the same time, it will limit the amount of choices available during selection thus reducing confusion. As an example, a designer may consider four relationships as they where defined in [Taleb et al. 2006; Javahery et al. 2007(b)] or six relationships summarized in [Todd et al. 2004].

Additionally, a pattern should contain an explicit definition of its *type*. More precisely, a set of patterns should be classified based on the domain of application. Some libraries [Welie, 2003] already organize their patterns based on the domain of application like Web, GUI and Mobile. Others [Tidwell, 2003], organize patterns based on the effect they have on the design: "Organizing the Page", "Commands and Actions" etc. There are also, new emerging libraries [Wilkins, 2003] that focus explicitly on a particular domain of application. Therefore, we believe that pattern types defined in terms of domain of application will allow the designers to easily classify and select patterns that belong to a particular domain or that are cross-domain applicable. Thus reducing the memory load on the designer and accelerating the design process.

At the same time, most of the patterns are aimed at facilitating a particular set of actions or tasks that a user may perform. Therefore, it is important to state explicitly which tasks these patterns facilitate. We have already mentioned that Tidwell [2005] has proposed a library that is organized into groups like "Commands and Actions", "Getting Input From

Users" etc. We notice that these names provide a subtle hint on types of tasks that these patterns are affecting. For example, "Command and Actions" refers to patterns that will influence the interaction during navigation. Similarly, "Getting Input from Users" contains patterns that may improve data entry and entry form usability. However, relying on the fact that a designer will read and take time to understand the headings for each group is insufficient. We propose to integrate this information as a variable of each pattern.

Overall, we propose to augment a commonly used pattern definition with the following: Short description, relationships, type and task. By clearly organizing patterns and establishing relationships between them, we will reduce interpretation and improve pattern selection. We will further detail each variable and propose a range of possible values for each of them in the coming chapters.

5.3 Linking persona to patterns

Creating a process that uses the best current practices and employs both patterns and personas seems to be a logical continuation of the previously surveyed state of the art. However, little work has been done in this area. Cooper and Reimann [2003] have proposed a Goal-Directed design and Welie et al. [1991], Welie and Traetteberg [2000] have proposed a Layered Model for Usability. While both proposals attempt to reduce the gap between user-experiences and design solutions, they do not clearly describe the process and cannot be directly used.

We propose to link the persona to patterns by assessing the effect of user needs and context of use on a particular design and selecting patterns according to the results obtained from assessment. For example, the simplest application of this idea would be to select web patterns only for a web site project. A more complex association could be in selecting a *wizard pattern* when a particular persona is presented as a *novice* user that needs *a lot of guidance* in a *sequential task.* In fact, designers tend to perform a logical analysis based on a sequence of assertions emerging from their experience that eventually leads to a particular pattern selection. Considering a design within boundaries of UCD in a case when persona construct is used, following knowledge was extracted:

1. Design decisions are dictated by expert's knowledge and interpretation of usability principles.
2. User needs can be divided into usability needs and special needs.

   - *Efficiency of use* is a usability need of *colorblind users* that have a special need- *avoid color encoding.*
   - A *special need* is defined as a characteristic resulting from a particular context situation. Fro example, various physical deficiencies may be seen as a special user needs. Similarly, low light working conditions may be in origin of a need for high contrast color schema. In general, a special need has a direct implication on a design.
   - In contrast, a *usability need* is a more general characteristic that may originate from a complex interaction of user characteristics and context of use. For example, it may be clear that a particular user requires high levels of guidance; however, this situation may occur because of previous experience, believes and other personal characteristics of the user the same as because of the need for high

security in working environment. In general, a usability need has an indirect impact on a design.

3. Usability needs have implications on physical design. To solve a need is to find a pattern that has the same implication on the design as the one suggested by a usability need.

   – For example, efficiency of use is a usability need that implies that a physical design will use methods accelerating data entry and function selection processes, such as *accelerators*. To satisfy this need we may select macros [Tidwell 2005] pattern that allows for execution of a set of prerecorded actions in a batch; thus, accelerating the process. In fact, *macros* pattern is an *accelerator*.

4. Special needs can be generally addressed by selecting a specific pattern.

   – In context description of redundant encoding pattern [Wilkins 2003], we find that it is applicable in a case when users may have visual deficiencies such as colorblindness.

5. During design, specialists build a mental model of candidate patterns, their relationships to user needs (eventually personas) and other patterns, and given context of use. Previous design experience then helps to make a decision and select a best suitable set of patterns.

Following the first affirmation concerning the usability principles, we have attempted to draw their comprehensive list from various proposals: Bastien and Scapin's Ergonomic Criteria [1993], Nielsen's Usability Heuristics [1991], Shneiderman's Golden Rules of Interface Design [1998], Van Duyne et al.'s Design principles [2003], and other sources [Padda 2003; Bevan and Macleod 1994; Bevan 1995; Bevan 1997]. This resulted in a set of 13 principles. However, we found that these principles could be further divided into two smaller groups: 1) principles closer to cognitive factors (see Table 11) and 2) principles closer to design factors (see Table 12). We have already seen an example of the accelerator that satisfies the need for efficiency of use. In this example, *efficiency of use* is a "cognitive" principle (let us call it usability need) while *accelerators* are a "design" principle (let us call them patterns or a design criteria).

Table 11: Usability Needs

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 the perceived means available to guide, advise, orient, inform, and instruct the user.
Minimal cognitive load	Is 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	Is the property, condition, or quality of being simple or un-combined (not composite). It often denotes perceived beauty, 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.

If we consider a design from the perception of a user, we may see that a user's usability needs are fulfilled by a set of pattern criteria. It is important to note here that a general user has no knowledge of performance; thus, *efficiency of use* will not be considered as a metric but rather as a purely cognitive impression, that user has of the actual performance. In fact, the performance may not change while the user becomes more satisfied with the design.

Table 12: Pattern Criteria

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 delivered to the user. Confirmation dialogues, prompting, status information, progress indicators, help, and documentation aids are a few examples.
Error Prevention	Refers to a set of methods applied in 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.
Grouping and structure (Logical Organization)	Refers to characteristics (such as location, format, behaviour, etc.) of objects and actions facilitating hierarchical, sequential and/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.

On the other hand, *pattern criteria* from the user's perception may be seen as actions, solutions and decisions that have a direct impact on the design. In fact, application of *pattern criteria* has a "tangible" or actually visible impact on the resulting design.

Since we found that surveyed usability principles could be farther divided into two groups, it is important now to explore the interactions between these two groups. Early literature analysis of various sources [Bastien and Scapin 1993; Shneiderman 1998; Nielsen 1994; Duyne et al. 2003; Welie et al. 1999] has already shown us that there were already few attempts to establish the interactions between different usability aspects, such as Welie' Layered Model of Usability (Figure 5).

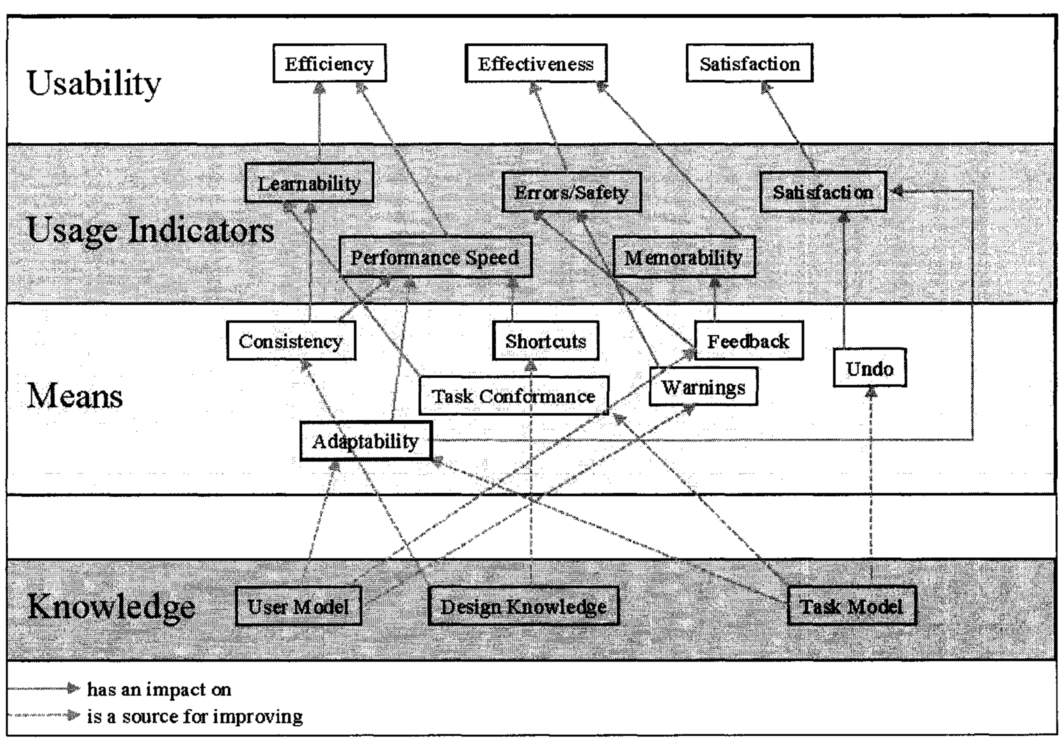


Figure 5: Layered Model of Usability [Welie et al. 1999]

In contrast with the figure above, we have attempted to establish dependencies between usability needs and pattern criteria. In other words, we have searched for design criteria that need to be considered to fulfill a specific user need. The general research approach was proposed by specialists in a focus group. Designers have intuitively decided to use concrete examples from their previous experiences to demonstrate each choice. This has given us the possibility to discuss between different groups of specialists their choices and come to a certain consensus. Unfortunately, specialists were unable to come to a consensus concerning the types of dependencies. The dependencies presented in the table below (Table 13) are undefined. We have extracted dependencies specifying that a designer needs to consider certain pattern criteria to fulfill a specific usability need. However, it is unclear how a variation of the strength of the need affects the importance of the given criteria.

Table 13: Dependencies between usability needs and pattern criteria

Pattern Criteria / Usab. Needs	Consistency	Error Handling	Error Prevention	Feedback	Grouping/ Structure	Min. Design	Navigation	Accelerators
Appeal					X	X		
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		

Similar to the above we have analyzed dependencies between *user variables* and *user needs*. Although all usability needs should be satisfied, some needs may become more critical for a particular user. Moreover, a design is a process that generally requires trade-offs to be made; therefore, it is a good idea to know which particular aspects of usability need may be sacrificed without major effect on users satisfaction. In the case of the dependencies between user variables and needs, we were able to establish a more detailed interaction model (see Table 14).

Table 14: Dependencies between user variables and usability needs

User Variable (x)	User variable domain (D)	Needs affected
Decreasing dependencies		
Education level	None, elementary, high school, vocational, college, undergrad, advanced	SL, CL, G
Linguistic ability	None to fluent	NM
Attitude to IT	Negative to positive	A, G, CL, S
Initiative taking	Reactive to proactive	G
Behaviour to features	feature shy to keen	CL, S, SL
Level of motivation	None to high	A
Age	Toddlers and children	A, SL, CL
Age	Adolescents	C, A
Computer Experience	None, Basic	G, S, SL, NM
Literacy	None, Basic	CL, NM, SL, S
Product Experience	None, Basic	G, S, SL, NM
Increasing dependencies		
Need for Guidance	None to high	G
Need for validation of decisions	None to high	G, S
Need for control	None to high	C
Initiative taking	Reactive to proactive	C
Age	Seniors and elderly	SL, G, S, CL
Product Experience	High, Expert	E, C

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

In fact, designers were able to specify which type of dependency exists between some pairs of variables. We have defined two types of dependencies increasing dependency and decreasing dependency. Similar to their mathematical counterparts (increasing function and decreasing function), increasing dependency is defined as $f(b) > f(a)$ and decreasing dependency is defined as $f(b) < f(a)$ for all $b > a$, where $a, b \in D$, $f(x)$ is a function of dependency between user variable x and user need calculated by applying the function and D is a domain over which x is defined and $f(x)$ exists. It is important to note that the rate of variation is unknown for all dependencies.

5.4 Summary

In this chapter, we have presented the requirements and gathered knowledge necessary to design or refine a process incorporating both user experiences (in form of persona) and patterns. Based on the previously accumulate knowledge from experts and case studies (NCBI), we have presented the requirements for user, persona and pattern format. We have also looked into the process of pattern selection as it is performed by designers.

More precisely, we have detailed user and persona unified models. In fact, we have defined a common model for both persona and all users used to build the persona. A section containing a set of discrete range variables has been added to a commonly accepted and used textual description. It has been proposed to use both narrative and discrete sections as alternative views on the same information.

Similarly, we have detailed a set of requirements necessary of constructing an effective pattern model. We have defined a small set of new attributes that we suggested to add to a commonly used model for patterns. However, we have not detailed the format of the pattern description on a premise that slight variations are possible depending on a process where these patterns are used.

Finally, we have documented rationale used by experts while determining pattern candidates based on user information. We have presented a set of dependencies between user variables and user needs, the same as between user needs and pattern criteria. In our opinion, it is a first step towards a systematic approach for pattern selection based on persona specifications.

6. P2P Process - Overview

Based on the initial knowledge extracted from the NCBI study and refined by information collected from focus groups and interviews, we have constructed a process following a natural flow as currently applied by the experts. In this chapter, we will describe the process following its simplest definition: "*a process is a set of actions transforming inputs into outputs.*" Therefore, we will present an overview of each phase of the process with inputs, outputs and key artefacts used. Throughout our description, we will follow a case study that will allow us to demonstrate the application of the explained theory.

Case Study

Our hypothetical example consists of a user interface design for a mobile application called crazy shopper. In this game, users can access virtual shops and with every shopping purchase receive points. Purchases include items categorized as clothing and fashion, household items, or electronics and gadgets. The goal is to be a "smart" shopper: the user must obtain a maximum amount of points in a certain timeframe while searching for items in all categories necessary for his living in the city. Frequent players with high scores receive gifts and shopping incentives from sponsors targeted according to their purchase habits in the game.

6.1 Process Description

Before we consider the steps comprising the process, let us establish the requirements. Our process requires a set of inputs to be present in order for an output to be produced.

More precisely, the process requires (1) user data, (2) pattern library and (3) context information to be present as inputs.

User data should be sufficient to accurately describe the population concerned. That is, the information collected previously should have enough details for the process to be applied. Similarly, pattern library should be relevant to the design problem at hand. For example, both web and visualization patterns are relevant for a web application allowing visualization of 3D molecular models. Finally, context information is essential for the key decision in the process. Results from empirical studies and other UCD artefacts allow the designer to gain an invaluable knowledge about the users, their needs and behaviours.

If all the requirements are satisfied, the process is expected to produce a conceptual design for a given user group in a given context. Moreover, it is expected that the process produces a set of intermediary artefacts documenting key decisions made. In fact, each phase of the process will produce a documented output that serves as an intermediary artefact.

Overall, the process is comprised of three phases: Persona creation, pattern selection and pattern composition (see Figure 6). Persona Creation takes as input user data and produces a set of representative personas. Pattern Selection takes as input the obtained personas and a pattern library, and produces an ordered set (based on importance and relevancy) of suggested patterns for the design. Based on the suggested patterns, Pattern Composition phase is expected to produce a conceptual design. It is important to note that context information serves an input during all phases of the design.

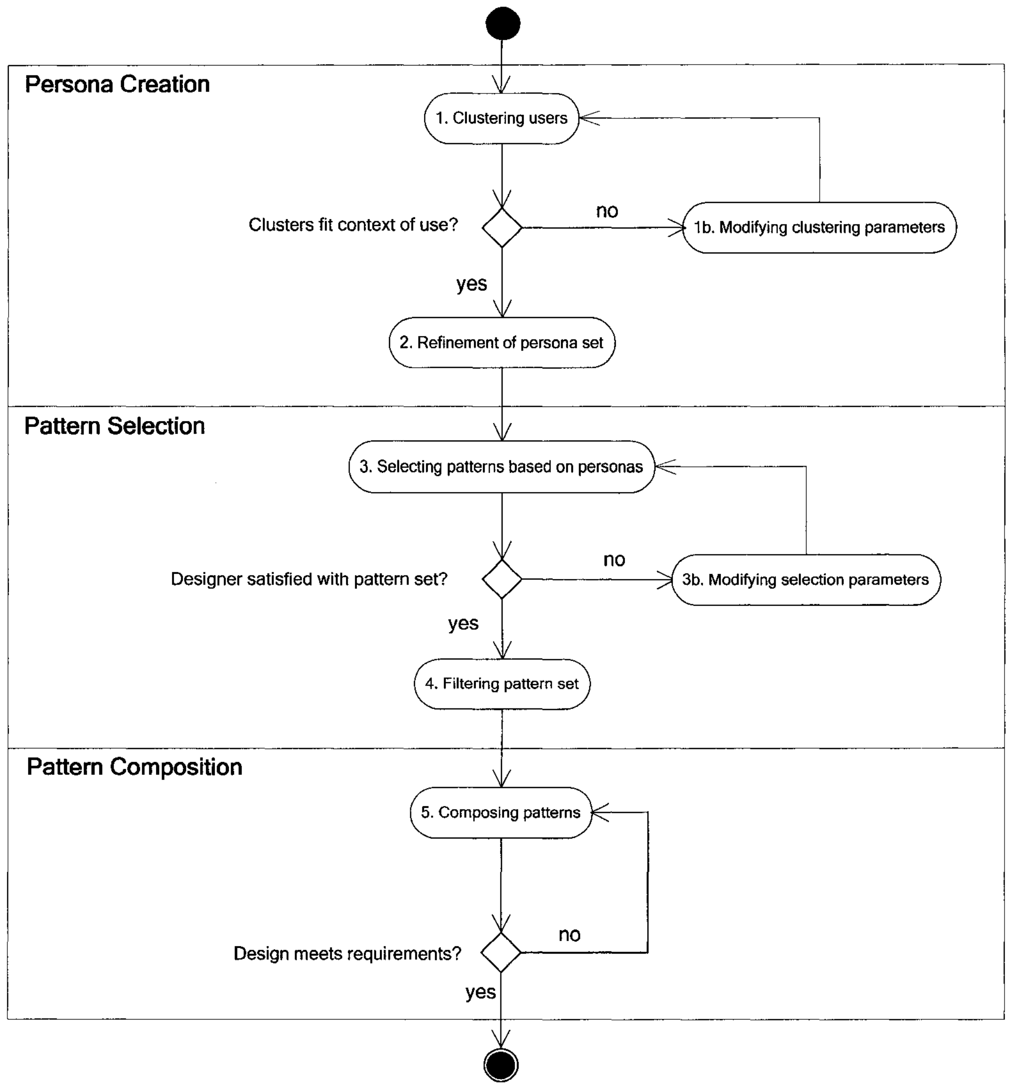


Figure 6: P2P Process Diagram

Persona Creation phase consists of three steps and a decision point: clustering users, verifying that the clusters fit the context, modifying clustering parameters if needed and refining personas. In essence, clustering consists of grouping users based on their similarities by analyzing a set of parameters. Once the users are grouped, it is important to ensure that the resulting clusters fit the context of the current design. If the verification proves clustering inadequate, the designer must modify the parameters used and repeat it.

In fact, clustering should be repeated as many times as required until the results are satisfactory. Once clusters are appropriate, each resulting cluster can be represented by a skeleton persona. Finally, these skeleton personas can be refined by completing the description based on real user data.

Once the information is reduced to a small and condensed set, pattern selection can be performed. More precisely, the designer selects patterns based on personas, evaluates the selected set and filters it. In order to select the patterns, the designer should use the output from the previous step, that is, personas. For each persona, patterns are ordered based on their importance using expert knowledge encoded as rules. The obtained set of rules is then evaluated based on the overall knowledge about a given persona, context of application and other relevant information. If the designer is not satisfied with the results, the selection parameters are modified and the last step is repeated. On the other hand, if the designer is satisfied with the proposed set of patterns, the set can be filtered based on the envisioned design model.

Finally, pattern composition phase requires the designer to combine patterns by employing pattern-oriented design techniques. This is a pure design phase, where the designer exploits all available knowledge from previous experiences and current project with information obtained from patterns (such as relationships, forces etc.).

6.2 Input Models: user, persona and pattern representation

As we have established earlier, the process requires a set of inputs, including user and patter descriptions. Therefore, before we can continue with the rest of the process we should define a format and describe the content of both descriptions. Since we hope to have a computer-assisted process in feature, both descriptions (user and pattern) should be computer and human readable.

By human readable description, we mean information that can be understood or interpreted by a human being, transmitted (written, spoken etc.) in a language commonly used for communication between humans. For example text, video and audio descriptions of the test performed with a user. On the other hand, a computer readable description should be encoded in a format that allows for a relatively easy manipulation in a computer-assisted environment. For example, binary or integer values representing sex and age of a user.

Based on the considerations above, we have produced a set of data formats describing the primary artefacts involved in a process. More precisely, each format has been defined as a set of variables defined over a certain domain. Additionally, we have also stated some transactional limitations involved.

6.2.1 User Variables

As it was described earlier, a good set should contain two characteristics: 1) be human readable and 2) be computer readable. Therefore, we propose to separate the variables into two sets: 1) augmented persona description, 2) discrete representation of that description. This separation allows distinguishing clearly between narrative information and details extracted from it. Such a distinction is essential in order to provide some form of explanation to the decisions made during the extraction of discrete variables. For example, during a process review a question may arise concerning a user classified as "proactive." In order to establish the source of the decision, the specialist will review a narrative description of user's behaviour with a video presenting a user during the test.

Now let us establish a definition of a narrative section of a user description; we consider that it should be short, clear and sufficient. As we saw earlier in section 2.3, personas descriptions have rapidly evolved and there is no clear standard available. Thus, we propose to use a model based on works of Pruitt and Grudin [2003] and Courage and Baxter [2005]. However, it is not yet clear whether such a description will fulfill the three criteria presented above: short, clear and sufficient. Moreover, further analysis shows that these criteria are impossible to measure, analyze or prove because we are considering a narrative text that allows for an infinite number of descriptions of the same object or phenomena depending on the writer's or reader's style, mood etc. Additionally, it is clear that the knowledge will evolve and change in the future in a way that could influence the interpretation of the narrative description. Therefore, for the purpose of our research and presented process, we defined a *user description as a current most commonly accepted and used format that, under normal conditions, should provide a description containing the information necessary in order to extract all discrete user variables*. For example, currently proposed description contains: (a) personal characteristics, (b) computer and domain specific skills, (c) interaction behaviour with the site or application, and (d) different tasks performed.

Now let us consider the second part of the user description - computer readable discrete user variables. We have already presented a set of variables that we have collected during our study (see 5.1). These variables have been defined with a goal to support automation. Therefore, they are perfect candidates for an initial set of discrete variables.

Once we have established a model, we need to identify a language that will describe it. Selected language, should allow for easy modification and evolution of the model it is used to describe. At the same time, it should facilitate the integration of the proposed model in a computer-aided process. Finally, we must consider currently popular and commonly used languages.

Following all the considerations presented above, we have used XML (eXtensible Markup Language) to define the user format. In fact, we have separated the user definitions into two categories: textual and discrete descriptions. Each category may have a sub category or actually an element holding some information. In the currently proposed format, textual description only holds elements containing narrative text, such as name, physical characteristics and others. On the other hand, discrete description contains a set of sub categories organizing elements into smaller logical groups following currently accepted classification. For example, discrete description contains a subcategory "Knowledge and Experience" that encapsulates elements such as "Education Level" and "Domain Experience." Moreover, each element in discrete section has been defined over a small range of values and maybe labeled as optional. For example, "Education Level" element, in "Knowledge and Experience" category, is defined over a set of nine values ranging from zero to eight and is optional to use. This allows us to encapsulate the knowledge about variables into XML, reducing the amount of references required to consult by an expert. User definition in form of XML schema can be found in Appendix E.A: User definition as XML Schema.

Case Study

Before we continue, let us see the above format applied in our hypothetical design of *crazy shopper application*. We assume that an appropriate set of domain and user studies was conducted to gather information concerning a potential set of users. Moreover, we have access to market segmentation information concerning mobile phones, but no specific information related to the same or similar application because it is a new concept. We also conducted a usability study with 25 potential users by administering questionnaires and observing interaction behaviour with a low-fidelity prototype. A set of potential users have been selected from current mobile phone users with subscriptions that include games. Each user is represented in a format following our model. Moreover, we have added one variable not originally present in the persona model – *IT acceptance*. We believe that IT acceptance depicts a different essential facet of user behaviour then IT attitude (already present in the model). The domain of the variable is {early adopter, average adopter, and technophobe}. Following is a description for user Margo (see Table 15). Note that comments include translations of numerical values for convenience.

Table 15: Description of Margo following proposed model

Simplified XML representation	Comments
<users> [...] <lname>unkown</lname> <fname>Margo</fname> <General Profile > Margo is a 17-year-old high school student, in her final year. She loves to find out what are the latest trends in fashion. She has a social life, although	 *A condensed narrative description of all important user characteristics.*

Simplified XML representation	Comments
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.	
</General Profile>	
<Physical characteristics >	
17 years old high school female student without any visible physical difficulties.	
</ Physical characteristics>	
<Cognitive style />	
<Goals />	
<Scenario />	
<Function />	
<Interaction details>	
Margo gets easily frustrated if system takes too long to respond.	
She likes to explore first before selecting an item to purchase19.	
She engages in short playing times, but frequently (daily).	
She engages in shopping sprees with her friends.	
</Interaction details>	
<Demographics>	
<Age>2</Age>	*Adolescent ,15-19 years*
<Income Level>1</Income Level>	*Somewhat low*
<Gender>1</Gender>	*Female*
</Demographics>	
<Knowledge_and_Experience>	
<Education Level>2</ Education Level>	*High school*
<Literacy>4</Literacy>	
<Computer Experience>2</Computer Experience>	*Average*
<Product Experience>3</Product Experience>	*Advanced with mobile games*
<Linguistic Ability>3</Linguistic Ability>	*Somewhat fluent*
<Background>false</Background>	
</ Knowledge_and_Experience>	
<Needs>	
<Control>3</Control>	*Somewhat high*
<Guidance>0</Guidance>	*None to low*
<Validation of Decisions>1</Validation of Decisions>	*Somewhat low*
</Needs>	
<Interaction_Style>	
<Features>2</Features>	*Average*
<Learning Speed>3</Learning Speed>	*Somewhat high*
</ Interaction_Style>	
<Attitude and Motivation>	
<IT attitude>3</IT attitude>	*Somewhat high*
<IT acceptance>0</IT acceptance>	*Early adopter*
<Motivation>3<Motivation>	*Somewhat high*
</Attitude and Motivation>	
</users>	

6.2.2 Pattern Variables

Now let us consider pattern variables. As it was established in the previous section, our set should satisfy the two basic characteristics of input and output: 1) be human readable and 2) be computer readable. Since we have already performed an analysis of user variables, we will reuse the same decisions here. More precisely, pattern description will be separated into two parts: 1) narrative and 2) discrete description.

Moreover, similar to the user description, narrative description of a pattern should be defined as a current most commonly accepted description format covering all aspects of the information carried by a pattern. A discrete description, on the other hand, is defined as a set of discrete domain variables facilitating and essential for mapping, selection and composition.

We follow Alexandrian format of "context, problem, solution" for narrative description of the pattern [Alexander et al., 1977]. Additionally, we include forces, examples and related patterns [Welie, 2003; Tidwell, 2005]. In order to facilitate the use of POD, a strong emphasis is placed on relationships between patterns. In an effort to accelerate the process of patterns selection when faced with a large set of them, we have added a *short description* attribute that combines the keywords from the context, problem, solution and relationships into a concise summary of essential information of the given pattern. As a result, narrative section contains a set of eight attributes: (1) Context, (2) Problem, (3) Forces, (4) Solution, (5) Examples, (6) Related patterns, (7) Short description, and (8) Relationships. Similar to a traditional use of patterns, a designer is expected to use these blocks of information in order to assess the applicability of the pattern for the current design, and examine the trade-offs and usability implications.

Following the definition, the discrete section should contain information required to adequately apply the process. More precisely, we need to extract information usually inferred by the designer, such as design criteria addressed by the given pattern. After analysis of a set of pattern libraries, we have found that information about design criteria, domain of application and particular user groups (if any) is embedded into narrative and requires careful reading. These attributes have been used as part of the discrete description.

Table 16: Discrete Pattern Variables

Pattern variables	Description	Values
Criteria*	A set of criteria related to the current pattern. In general, a set of two most important criteria that this pattern affects: primary and secondary.	Shortcuts/accelerators, Feedback, Error Prevention, Error Handling, Grouping & Structure, Navigation, Consistency, Minimalist Design
Pattern type*	Refers to a type or a section of the library this pattern belongs to. Usually, it is defined as domain of application.	Web, GUI, mobile, visualization
Task type	Refers to a generic description of task where a given pattern maybe applied to improve usability.	Unbounded set of descriptions based on a standardized format.
Relationships	A set of relationships with destination node specification. Contains information regarding relationships that current pattern has with other patterns.	Unbounded array of (relationship, destination pattern) sets. Relationships are defined in following sections.
Special needs	Refers to a specific group or a set of specific groups of users that this pattern is targeted for. In general, special groups are defined based on some special needs that these users have.	Colorblind, low vision, no vision, hearing disability, physical/motor disability, learning/cognitive disability, children, seniors, novice, expert, low literacy

* - required

Similar to user description, we have defined patterns by using an XML schema containing the information about range and type of the variables (see Appendix E.B: Pattern definition as XML Schema). However, following the process (described below) we have limited some of the categories to a small set of instances. For example, a pattern cannot affect more than two criteria. In fact, most patterns affect all criteria but to a different degree. In order to reduce the complexity, at most two most important criteria will appear in pattern description. First criteria must be seen as an aspect of a design that is directly affected by a pattern; a pattern has the strongest effect on this criteria. Second criteria must be considered as secondary with partial impact on the design. When compared to the first criteria, second criteria may be seen as the one having "half" of the impact of the first. Following this definition, it is possible that a pattern has only one criterion (first or second) (see Appendix F: Patterns and related pattern criteria).

Similar to a situation with design criteria, a large set of patterns belongs to all application domains. For example, Tree Table pattern [Tidwell, 2005] can be applied in web and standard GUI applications. However, in order to accurately classify this pattern, we have to consider the primary or most commonly used domain of application (in our case GUI). In the case when a pattern is *equally* used in multiple domains, all domains must be specified.

In previous sections, we have described our vision of *relationships* attribute. As we can see, in the table above (see Table 16) a relationship communicates three types of information: (1) source pattern, (2) destination pattern and (3) type of relationship between source and destination. For the purpose of the proposed process, we will define four types of relationships following a model proposed in UPADE [2004] and described in [Taleb et al. 2006; Javahery et al. 2006; Javahery et al. 2007(b)]: super-ordinate, sub-ordinate, neighbouring, and competitor.

1. **X Similar Y**: if and only if X and Y can be replaced by each other in a certain composition, then X is similar to Y and Y is similar to X. In other words, X and Y

are similar if each of them provides a solution to the same problem in a similar context. For example, index browsing and menu bar patterns are similar. They both provide navigational support in the context of a medium size website, allowing users to navigate among 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.

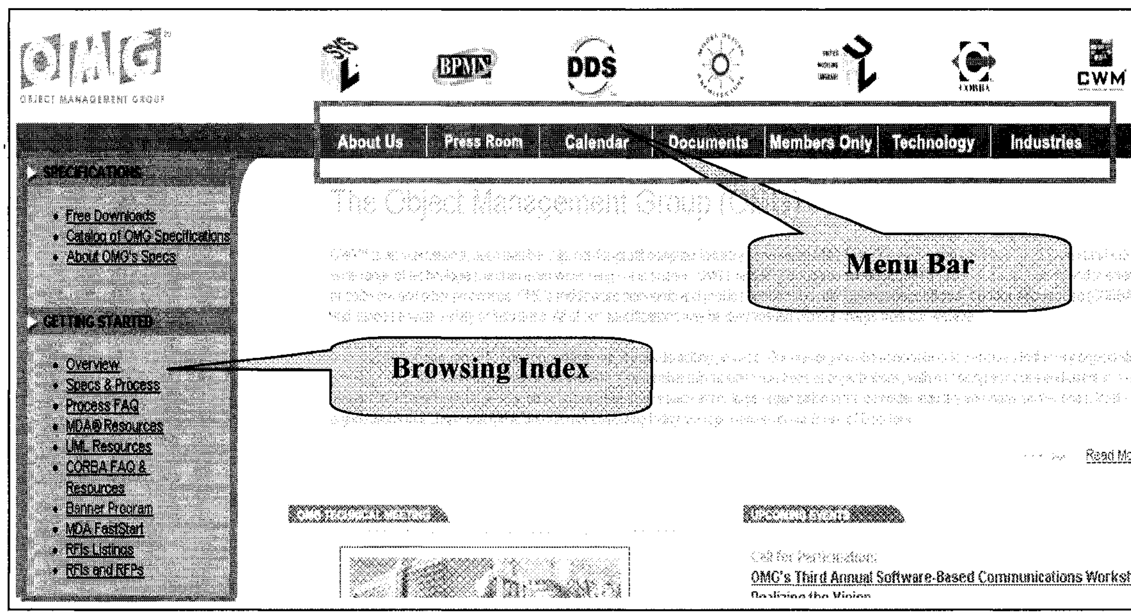


Figure 7: Similar Patterns example

2. **X Competitor Y** : if X and Y cannot be used at the same time, X is competitor to Y and Y is competitor to X. In other words, if X and Y are similar and the solutions they provide are equivalent (patterns are interchangeable), they are competitors. For example, the Web convenient toolbar and menu bar patterns are competitors. The convenient toolbar solution states: "<u>Group</u> the<u> most common convenient action links</u>, 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." Despite small differences in the descriptions, both solutions should have the same content: Contain a group or a collection of most frequently used links. Therefore, both patterns are competitors. In the example below (see Figure 8), we notice that *Contacts & Feedback* is located in the Convenient Toolbar, while *About Cisco* is located in the Menu bar. In general, we would expect to have this information in the "same" place. This clearly demonstrates that both patterns are competitors and should not be used together to avoid confusion.

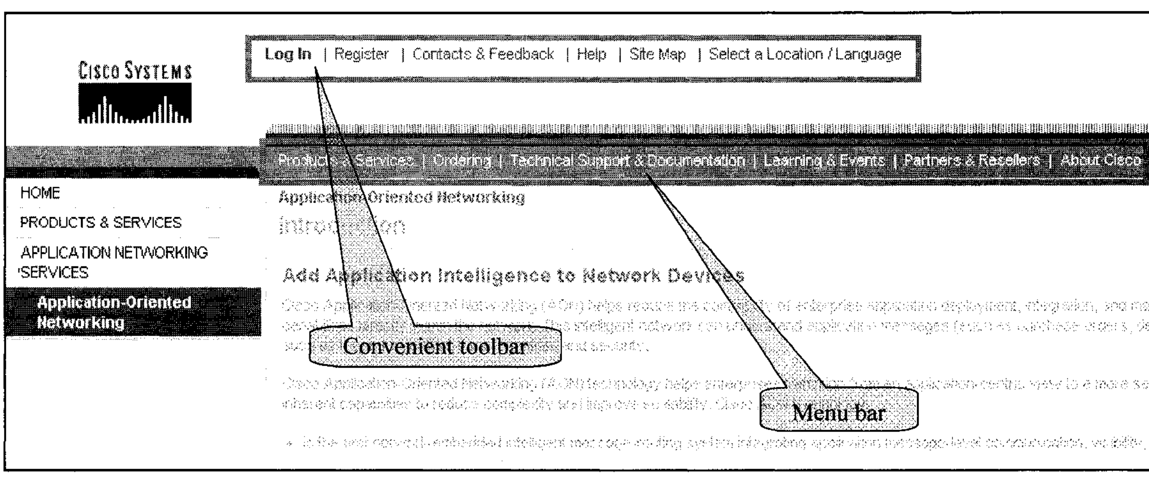


Figure 8: Competitor Patterns example

3. **X Super-ordinate Y**: if and only if X is composed of or uses a set of patterns including Y, then X is super-ordinate of Y. In other words, 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, Convenient Toolbar pattern may include Safe Places pattern (see Figure 9).

4. **X Sub-ordinate Y**: if and only if X is embeddable in Y, then X is sub-ordinate of Y and Y is super-ordinate of X. This relationship is the inverse of a super-ordinate relationship. For example, the convenient toolbar is a sub-ordinate of the home page pattern (see Figure 9).

5. **X Neighboring Y**: if X and Y belong to the same pattern category (family), then X and Y are neighboring. For example, Index Browsing and Convenient Toolbar patterns are neighbouring because they both belong to category of Navigational Patterns.

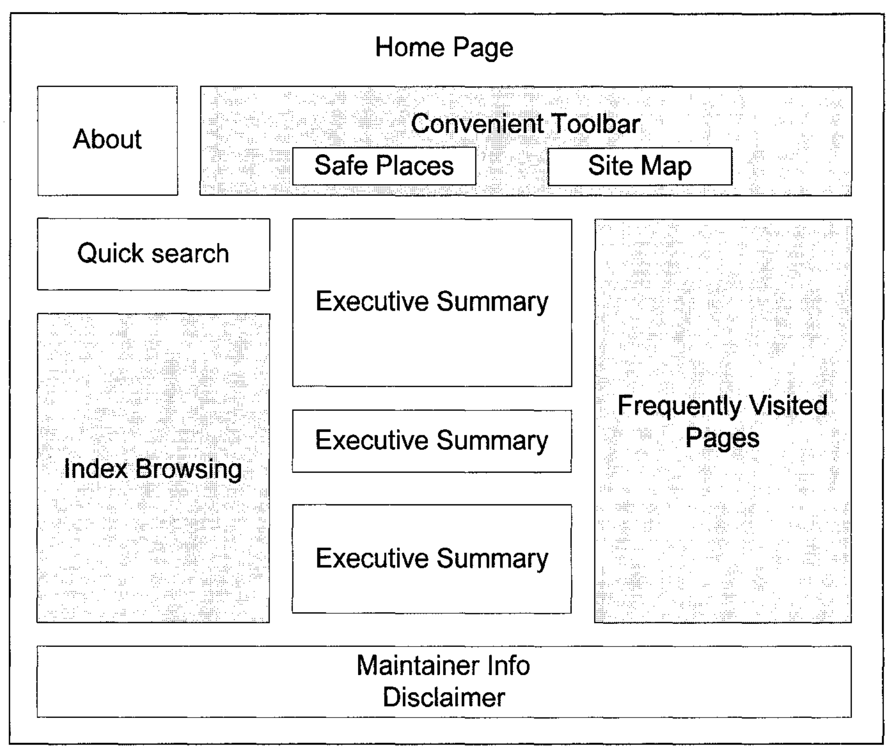


Figure 9: Composition of patterns

We find that the relationships proposed above form a solid base for future development and improvement. With available relationships, a designer can rapidly navigate through patterns that are complementary, incompatible or alternative to a given design. Therefore, a designer can relatively rapidly analyze all possible combinations and construct a list of alternative solutions. We will demonstrate the application of relationships in the Pattern Composition section (7.3).

Case Study

As an example of a pattern description, let us consider a reduced pattern description that maybe useful in our crazy shopper application. A Legend pattern [Wilkins, 2003] is included in the pattern library proposed for a case study as a *Visualization* pattern. It maybe used when there is a need to interpret a schema or any other information encoded in form of symbols.

We may note that in this example (see Table 17) some information is repeated twice. For example, we have a section that describes a pattern type in both sections (discreet and text variables). Similarly, narrative description contains a section called Related Patterns, while discrete section contains an attribute called Relationships. This redundancy is due to one of the requirements that we have previously defined. A textual description must contain all the information necessary to extract and explain the values contained in the discrete, computer readable section.

Table 17: Legend Pattern following proposed model

Simplified XML representation	Comments
<patterns <Title>Legend</Title> <Short Description> Use legends as a memory aid to assist the user in the interpretation of the display. </Short Description>	*Minimum of information to understand what this pattern does.*
<Context> Data is encoded in a form of visual objects that are then organized in a scene. </Context> <Problem> How to remind the user how to interpret the display. </Problem> <Forces> Visual objects are displayed that encode multiple data attributes. [...] </Forces> <Solutions>Use legends as a memory aid to assist the user in the interpretation of the display. Legends and annotations help to reduce some of the cognitive load on the user. Datatips is a good annotation technique that only requires attentive viewing when the user requires it and only temporarily increases clutter. [...] Labels in close proximity to objects also help to increase identification accuracy but at the potential cost of more attentive viewing and an increase in visual clutter. </Solutions> <Examples> SeeIt, SP500 Minimalist Representation (Brath 1999); Urban View (Salisbury 2001); any standard char-based presentation e.g. bar chart, scatter plot. </Examples> [...]	*A detailed description including examples and all other relevant information.*
<Related patterns>none</Related patterns> <Type>Visualization pattern</Type>	*Information helping to organize and classify patterns.*
<Relationships /> <Information> <primary criteria>Grouping_and_Structure</primary criteria> <secondary criteria>Minimalist_Design</secondary criteria> <type>visualization</type> <special /> </Information> </patterns>	*Discrete variables, used during selection and mapping.*

7. P2P Process - Detailed phase description

We have already presented three key phases of the process including manipulated artefacts, such as patterns and persona. However, the process description was at relatively high level of abstraction and did not offer sufficient details for implementation of the P2P process in practice. In this chapter, we propose to follow a detailed description of each phase of the process. Moreover, we will continue our sample case study that we have started in the previous section.

7.1 Persona Creation

Persona Creation phase comprises three major steps Clustering, Modifying Clustering Parameters and Refining Persona Set. Recall that following the process description, we assume that all user information have been already compiled in appropriate format. Therefore, the first step of persona creation consists of manipulation of user data. More precisely, a designer is called to group users based on asset of parameters described below. Then a designer is called to evaluate a resulting persona set and is given two choices: to continue the process or to refine the parameters and repeat the first step. If the resulting persona set is found suitable the designer proceeds to the last step, Refining Persona Set.

7.1.1 Clustering

Clustering is a statistical comparison technique aimed at dividing data points into a set of homogeneous groups containing data items as close to each other as possible. At the same time, clustering is some times considered as a form of reduction of a large set of data into a smaller set of representative prototypes or clusters. Depending on data and an application, different types of similarity measures may be used to identify classes, where the similarity measure controls how the clusters are formed. Some examples of values that can be used as similarity measures include distance, connectivity, and intensity.

In non-fuzzy or hard clustering, data is divided into "crisp" clusters, where each data point belongs to exactly one cluster. In fuzzy clustering, the data points can belong to more than one cluster, and be associated with each of the points by membership grades, which indicate the degree to which the data points belong to the different clusters.

During our data elicitation process, we have discovered that the designers intuitively apply a fuzzy clustering technique while considering only a subset of user variables. More precisely, the designer initiates an iterative process while considering a set of related user variables and the groupings of users within and between the variables.

For the purpose of the proposed process, clustering has been detailed as follows:

1. Select influential user variables based on variations in interaction behaviour 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.

   - *Prioritization should be done based on the observed variations in behaviour and needs within given context.*
   - *It is preferable to select only two, three variables at this stage and allow for refinement during Modifying Clustering Parameters step.*

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. Cluster users by applying most appropriate technique. In order to further detail the possibilities, we propose the following technique:

   4.1. For the first variable, determine the users that fall under each value. Each set of users results in a cluster.

   - *If the sample size is large, percentage of users can be used.*
   - *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 important in terms of the design context, the restricted domain can be extended to create a new group of users with this value.*

   4.2. Dependencies between variables should be considered. A dependency results when a combination of variables causes some interaction behaviour, but this behaviour does not result when each variable is considered in isolation.

   - *This may result in multiple variables being considered together for clustering.*
   - *For example, if the combination of age group and IT acceptance causes a significant behavioural change, we need to preserve this information (see example below).*

4.3. Clusters need to be analyzed by considering their importance to the design context. Affinity diagrams can be used if the clusters are too numerous to handle.

  - *Consider the information that given cluster brings to the designer in relation with other available clusters. If this information is already included in another cluster, this cluster may be removed.*
  - *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 that may be important for the design context.*

4.4. For the next user variable, take the set of clusters and break them into sub-categories following steps 4.1-4.4. Repeat until you have created clusters based on all user variables in your selected list.

5. 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.
  - *Populate the variables that were used to create the final set of clusters with values from the given cluster.*
  - *Populate the rest of the variables with the most commonly occurring values in a given cluster.*

Following the steps described above, we have assumed that "user understanding" phase have been completed previously. In fact, the first step in Clustering considers that the designer knows the context and is capable of extracting and prioritizing the knowledge obtained during various domain analysis and field studies.

Let us now return to our case study about crazy shopper application. As we have noted earlier, let us assume that we have gathered all required information from potential users.

We are fortunate to have access to segmentation information for mobile phone use. Unfortunately, no specific information relative to games and other applications is available. Finally, we have conducted a set of usability studies with 20 potential users.

Case Study

1. Based on variations in interaction behaviour and needs we have selected *Age*, *IT acceptance* and *Gender*.
2. Based on our knowledge of user needs and behaviour we consider that *Age* and *IT acceptance* will be the two user variable we will use for clustering.
3. Now we will consider the range for both variables. In general, combination of these variables allows for 27 unique clusters.
    a. *Age* variable includes *toddlers*, *children* and *elderly* that are rare users of mobile phones. Moreover, we find that seniors and mature adults are not part of the target audience for our game.
    b. *IT acceptance* variable includes technophobes as one of the possible values. However, we know that these individuals are using mobile phones (10% of the market) but not mobile games ("insignificant share").
    c. As a result, range for both Age and IT acceptance is greatly reduced: {adolescents, young adults} and {early adopter, average adopter} respectively. We have gone from 27 to 4 possible unique clusters.
4. We will now attempt to break users into 4 groups based on the previously selected variables and their possible range. The result is:

Case Studv

User Variable		Cluster	Decision
Age	It Acceptance		
Adolescents	Early adopter	**C1**	Keep
	Average adopter	C2	Remove
Young Adults	Early adopter	C3	Remove
	Average adopter	**C4**	Keep

Moreover, we find that there is a subtle difference between early adopters and average adopters. We also find that there is no clear dependency between both variables.

Since our goal is to preserve information representing differences in interaction behaviour, we omit C2 and C3. In fact, by designing for C1 and C4 we can also satisfy the needs of C2 and C3. In other words, if the needs and behaviour of C2 and C3 are included in the set of needs and behaviours of C1 and C4, by designing for the latter we will be automatically designing for the earlier. Note however, that we could of selected C1 and C4 as the two representative clusters. However, market segmentation studies have shown us that users in C4 groups are more often present on the market then those in C3 group. Therefore, our finial clustering decision was made based on the real user sample.

7.1.2 Modifying Clustering Parameters

In the previous section, we have described a process that outputs a set of user clusters that then can be represented by personas. However, we have omitted an important step: we have not specified when and how a designer can evaluate the resulting clusters against all the information gathered from previous studies. In fact, *Modifying Clustering Parameters* is an optional sub-step of *Persona Creation*. Need for this step is evaluated by a designer when considering the resulting clusters.

Before performing *Modifying Clustering Parameters*, a designer needs to evaluate the coherence of the resulting clusters considering already available or newly obtained data. In fact, it is possible that the variation of the behaviour and/or needs within clusters cannot be explained by simple statistical error. In addition, it is possible that two or more distinctive sub groups emerge within a particular cluster. Further usability studies may be performed at this step bringing additional essential knowledge into the context.

Once it is established that current clustering results are not sufficient or adequate, the designer should revise the clustering performed earlier considering newly obtained information.

In the case of the crazy shopper application, we have performed a series of short studies with a low-fidelity prototype. The results have demonstrated a "clear" segregation within clusters. Further investigation has demonstrated that behaviour and needs of the users varied with gender. In fact, females showed a greater interest in playing the game and exploring virtual rooms, while males had a strong tendency to explore construction hardware, electronics and other gadgets. Moreover, this addition does not come as a surprise: during first clustering, we have selected three variables and *Gender* was one of them. Although we have removed it in the first iteration, we find that it is important and use it in the second iteration.

Case Study

During evaluation and analysis, it has been found that previously obtained clusters can be further subdivided based on the *Gender* of the users. Second iteration of clustering resulted in eight clusters:

User Variable			Cluster	Decision
Age	It Acceptance	Gender		
Adolescents	Early adopter	Female	**C1**	Keep
		Male	**C2**	Keep
	Average adopter	Female	C3	Remove
		Male	C4	Remove
Young Adults	Early adopter	Female	C5	Remove
		Male	C6	Remove
	Average adopter	Female	**C7**	Keep
		Male	C8	Remove

Now we will reduce the amount of clusters similar to the process already performed earlier. The knowledge about users leads us to a conclusion that we should keep C1, C2 and C7. Our decision is based on common behaviour of some clusters and the amount of users they represent.

Finally, we re-evaluate the resulting clusters and find that further subdivision does not add any valuable information.

Now we can create the skeleton user representing each of the clusters. This process is more of a statistical analysis than real creativity: we will use the most commonly available values in the cluster. For the cluster C1 we obtain:

User Variable	Value	User Variable	Value
Age	Adolescent	Income level	Low
Gender	Female	Computer experience	Average
IT acceptance	Early adopter	Education level	College
Linguistic ability	Fluent	Control	High
...	...	...	...

Let us now analyze the decisions made in the case study in order to better understand the process. We have to remind that the first pass of clustering resulted in two groups: *adolescent early adopter* and *young adult average adopter*. After the second pass, clusters have been changed and detailed farther. However, selected clusters preserved the information we have kept in the first pass: we have *adolescent early adopter female*, *adolescent early adopter male* and *young adult average adopter female*. During second pass, we have divided one cluster into two and removed a part of the second cluster. Since we know that, each cluster contains users that share common behaviour we are sure that by removing part of the second cluster from the first clustering pass we have not lost any information.

Now let us see if we have gained any new valuable information. In fact, greater granularity allows for a more detailed look into behaviours of each group. For example, young adult females spend more time purchasing household items, while adolescent females spend more time purchasing clothing and fashion items. This variation in behaviour is considered by the designers as essential information that will allow them better design and target the game for each group of users.

The last step is to re-evaluate the resulting clusters in order to consider if further clustering is necessary. At this step, the designers decide that further iterations are not necessary. This decision is based on analysis of behaviour and at the same time is strongly dependent on the insight of the designers and their expertise. As with any iterative and creative activity, the decision to stop or continue clustering iterations should not be solely based on statistical analysis it must also reside on the designer's knowledge of the field, users and domain of application. However, if time or resources (such as tool support) allow it, many iteration paths can be attempted and a designer may revert or select the path that has lead to the best results. In our case, the designers have decided to stop at three clusters:

- C1 - adolescent, early adopter, female
- C2 - adolescent, early adopter, male
- C7 - young adult, average adopter, female

7.1.3 Refinement of Persona Set

In the previous sections, we have analyzed the process necessary for dividing users into clusters. We have seen how we can divide users into groups based on information obtained from market and usability studies. The output of clustering was a set of clusters and their representatives. In other words, for each cluster, we constructed a representative user that contained values for variables we have used for clustering and other most commonly occurring values from the cluster. However, we have to remember that persona is a communicative tool. It should be engaging and should contain a description that will be sufficiently detailed to imagine a given person in the scope of the project. Therefore, a skeleton constructed earlier must be modified and refined to contain all information: textual descriptions, picture, name and other details. We suggest following steps:

1. Create a personage, like in a fiction book. Character traits, habits, working environment details etc. should be added following persona model described earlier.

- *To do this the designer may use the information from real user belonging to the cluster.*
- *The designer must use representative values and information from the given cluster.*
- *The designer must pay particular attention to the interaction behaviour of the emerging persona. The resulting description should not contain characteristics that have not been used in clustering but that may considerably change user's behaviour.*

2. Create an identity. Name (first name only or a full name can be used); picture and personal quotes are essential components that add a sparkle of life to the persona.
3. Re-evaluate resulting persona.
   - 3.1. Verify that the values correspond to textual description.
   - 3.2. Adjust the values when needed.
   - 3.3. Verify that resulting persona is representative of the corresponding cluster.
   - 3.4. Repeat steps one and two if needed.
4. Repeat previous steps for all clusters.

As usual, let us now practice on the crazy shopper case study the proposed sub-process. We have stopped at creating three skeleton users that consisted of the more commonly present values in our clusters.

Case Study

Following the skeleton described in the previous section we will create a persona representing cluster C1.

We know that this persona is a female adolescent and an early adopter. Moreover, we can use some insight given by a skeleton persona. For example, education and income suggest that this persona will be a student. Aggregation of the information found in the cluster and skeleton of the persona yields:

- 18 years-old college student.
- Loves shopping and going out with her friends.
- Uses mobile phone to keep essential information in it, like contact numbers, important events in calendar, and play games.
- Likes to try new games as often as possible.
- […]

Now we can create an identity for this persona:

Her name is Anna Spinelli. She loves Latin and tries as much as possible to follow the advice of Horace: *dum loquimur, fugerit invidia aetas: carpe diem quam minimum credual postero*. That is, *even as we speak envious time is running a way from us: gather the day (seize the day), for in the future you can believe the minimum.*

It is time now to verify that our persona represents its users. When comparing values we find an interesting trait that was missed earlier. In fact, Anna, similar to the users she represents, has a strong need for control and efficiency and at the same time has a strong need for exploration. While these characteristics are new and may affect the design decisions, we should keep them because they appropriately reflect the characteristics of real users.

Now we have obtained the first persona (see Table 18 adopted from [Javahery 2007]) representing one of our clusters. We should repeat the process to create the other two personas before moving to the next step.

Table 18: Narrative description of Anna (persona for C1)

<table>
<tr><td colspan="3">Anna Spinelli</td></tr>
<tr><td></td><td colspan="2">"Dum loquimur, fugerit invidia aetas: carpe diem quam minimum credual postero."

That is,
"even as we speak envious time is running a way from us: gather the day (seize the day), for in the future you can believe the minimum"
- Horace</td></tr>
<tr><td>General Profile</td><td colspan="2">Anna is an 18-year-old college student. She lives with her parents. She is in her first year of the Arts and Languages 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 important events in the calendar, and likes playing games on it.</td></tr>
<tr><td>Goals</td><td colspan="2">Professional: Succeed in school, and work towards an university degree in Arts and Literature. Personal: Enjoy time with her friends and family. Application: Have fun for a few minutes, and then get back to studying.</td></tr>
<tr><td rowspan="4">Scenario</td><td colspan="2">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 crazy shopper game. 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.</td></tr>
<tr><td>Specific needs</td><td>Control, Efficiency, Exploration</td></tr>
<tr><td>Features</td><td>Quick loading, scoring recall/tracking, rapid-key exploring, silent mode (visual indicators)</td></tr>
<tr><td>Interaction details</td><td>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).</td></tr>
<tr><td>Demographics</td><td colspan="2">She is an 18-year-old female, with a student income and financial support from parents. She has a part-time job allowing her to pay for social and personal expenses not related to school.</td></tr>
<tr><td>Knowledge and Experience</td><td colspan="2">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 crazy shopper a few months ago.</td></tr>
<tr><td>Psychological profile and needs</td><td colspan="2">She needs to be in control, and is a fast learner (high learning speed). She needs basic (low) guidance and no validation of decisions.</td></tr>
<tr><td>Attitude and Motivation</td><td colspan="2">She has a positive attitude to IT, and somewhat of a high level of motivation to use the system.</td></tr>
<tr><td>Special Needs</td><td colspan="2">She has no disabilities or any other special needs.</td></tr>
</table>

7.2 Pattern Selection

The second milestone is the Pattern Selection consisting of logically linking a set of patterns to a set of personas obtained by clustering. The Pattern Selection method involves three steps: 1) Selecting patterns based on persona 2) Modifying selection parameters and 2) Task based filtering of patterns. Each step ads or subtracts information obtained from the previous step.

In the chapter 5.3, we have described expert knowledge in form of dependencies linking different levels of concepts that can eventually link user variables with patterns. We have proposed to encode expert knowledge in form of rules connecting different levels of layered model, based on initial idea from [Welie et al. 1999]. We propose now to design a rule-based technique that will utilize the expert knowledge to recommend a set of patterns based on the values of user variables.

Our goal is to design a process that will recommend or suggest a set of patterns based on some measure or logical deduction. Our goal is not to enforce a particular choice or to provide only one best-fit solution; it is rather to guide a creative design process. We have to remember that a design is a creative process that requires tradeoffs and other creative decisions to be made. Therefore, the designer should not be restricted in the selection of components (in our case patterns) during design. However, ordering of patterns based on their relevance for a given persona will greatly reduce the memory load and time wasted in search of patterns. Similarly, such a list will help in selecting the most applicable pattern from a list of a few competitor patterns.

Additionally, similar to a specialist intuitive thinking process, we need to link our set of patterns to the environment. We need to ensure that the patterns we have selected are applicable to the given domain and a set of typical tasks.

Given all the above consideration, Pattern Selection should consist of establishing a relation between a set or a single user variable and a pattern. Moreover, such a link should be established using a set of learned or predetermined logical operations. Such operations can take any form from a simple condition with a yes/no answer to a complex neural network with pre-learned weights.

We propose to use scoring technique to compute the relevance of a given pattern for a particular persona. In fact, we have adopted an idea taken from a recommender system [Kim and Kim 2003], where suggestions are made based on a computation of the confidence of the result (or as we call it score). More precisely, we define a score as a relative measure of relevancy of a given pattern in relation with other patterns in the same list, computed from the sum of scores. Therefore, the actual value of the score has no particular meaning. It is only in relation with other pattern scores that we can determine the relevance of the given pattern.

7.2.1 Assumptions

Pattern Selection sub- process is designed based on a set of assumptions:

- a set of available patterns is suitable for the given project and pool of users,
- clustering output presents a set of small clearly distinct groups of users,
- and the pattern selection process itself is flexible and suitable for the given application.

Let us consider each assumption in order to see what it means and what can be done to reduce its effects to minimum. We have assumed that a set of patterns suitable for the project at hand is available. More precisely, as it was presented earlier many pattern libraries exist today and all of them have taken a more or less different approach to patterns. That means that certain collections maybe more or less suitable for a given project. For example, we will attempt to use a visualization pattern library in a web site project. In such a case, two possible outcomes can be predicted: 1) there are no relevant matches produced and 2) there are some matches produced but none of them are relevant to the type of application. The first outcome will clearly indicate to the user that the process failed and will force to search for a solution or abandon the process. On the other hand, the second outcome produces a set of solution that can be found useless based only on the experts opinion and may not be easily detectable.

One of the possible solutions is a taxonomy based on domain of application. More precisely, it can be proposed to have patterns organized in a way that would allow for selection of subgroups of them based on their possible applicability to a current domain. Unfortunately, currently available pattern libraries do not have such an organization. Therefore, we conclude that Pattern Selection process should be lightweight or automated to allow for multiple iterations with different pattern libraries.

Our second assumption deals with the quality of the input. More precisely, we are considering the personas that have been obtained from clustering. Multiple errors may originate from the fact that constructed personas are not representative of the users. At the current stage, we have found no other way but to rely on the quality of the output from clustering. We must note here that we are not considering overlapping personas as a particular problem that may affect Pattern Selection process. As long as personas effectively represent real users, Pattern Selection process will produce corresponding and relevant results just for a larger set of persona.

In our last assumption, we are considering that the process itself is suitable for the project. For example, new developments in HCI and better understanding of the human behaviour may modify the ways we describe and see patterns and personas. In that case, the process and the models proposed below may need to be modified and adopted to the state of the art. Therefore, the process must always remain in sync with the format and type of inputs it receives.

Given all the assumptions presented above and the expert knowledge described in previous chapters, we can now consider the details of the proposed Pattern Selection sub-process.

7.2.2 Pattern Selection as a scoring technique

Scoring is a vague concept with a definition changing from field to field. It is commonly used to reduce errors during medical diagnosis. It is also applied in the gaming industry to identify the best player or a team. Overall, scoring consists of adding a predefined score to the one already available based on a rule triggered by an event.

We have followed the same strategy in order to suggest a set of patterns. In fact, a score in our process represents a confidence or relevance criteria that allows us to order suggested patterns. Moreover, the final score consists of a value attributed to each pattern after all relevant rules have been executed.

7.2.2.1 Rules

In general, a rule is defined as an association between two variables, where an association is defined as some known or unknown dependency between two variables. Recall that in the section 5.3 we have presented dependencies between usability needs and pattern criteria as unknown because the effect of one variable to another was unclear. On the other hand, dependencies between user variables and usability needs have been described in more detail: variations and ranges have been defined.

However, in the case of known dependencies the rate of variation has not been defined. In order to apply the proposed method of computation we need to approximate the variation rate. In lack of more detailed information, linear approximation is the only suitable option. On the other hand, it is possible to farther detail some of the variations (see Figure 7). For example, in a Table 14 presented in section 5.3 there is a set of associations between Age variable and Appeal usability need. Therefore, we can define it as a composite association, which is in fact a combination of simpler associations.

Similarly, a maximum association is used when a range of user values is associated with a single usability need value.

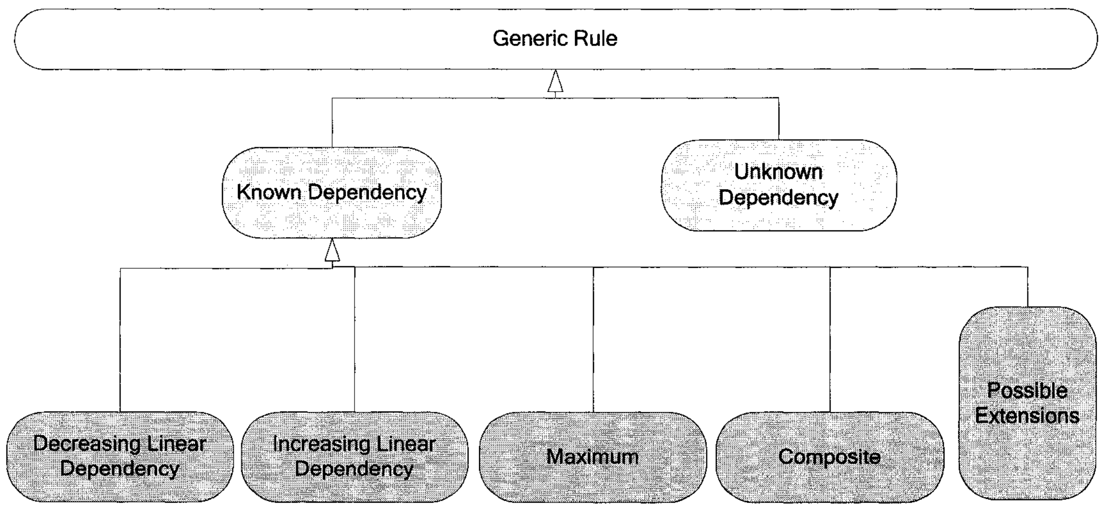


Figure 10: Rule types

Although we have defined only four relations, there is no limit on the amount or type of relations available, as long as the relation can be defined by a single or a set of simple mathematical (including logical) operations.

Before we continue let us define each particular dependency. In our definition, we will assume that a particular user variable $u_{i,j}$ belongs to the matrix of all user variables and is identified by its row (i) and column (j) values. Moreover, we will consider only cases that simply demonstrate the concept of the particular dependency without an attempt to provide a general rule for its definition. A more generic definition will be presented when we will evaluate the interactions between the rules.

Increasing Linear Dependency

An increasing linear dependency $r_{i,j,m}$ is a relation between a user variable $u_{i,j}$ and a usability need n_m. Moreover, the source of $r_{i,j,m}$ is a variable $u_{i,j}$ and a destination is a need n_m. Assuming that $u_{i,j} \in \{1, 2, 3, 4, 5\}$ and $n_m \in [0, 1]$, we can conclude that the value of n_m can be computed as follows:

$$N_m = \frac{\max(n_m)}{range(u_{i,j})} \times U_{i,j} = 0.25 \times U_{i,j},$$

where N_m is particular value of variable n_m, $U_{i,j}$ is a particular value of variable $u_{i,j}$, $\max(n_m)$ is assumed to return the maximum value that n_m can have and $range(u_{i,j})$ is assumed to computed the difference between the maximum and minimum values that $u_{i,j}$ can take. That means that the greater the value of $u_{i,j}$ the greater the value of n_m will be.

Decreasing Linear Dependency

Let us consider another example: decreasing linear dependency $r'_{i,j,m}$. Similar to the above, we have: the source variable is $u_{i,j}$, the destination variable is n_m, and values are defined as $u_{i,j} \in \{1, 2, 3, 4, 5\}, n_m \in [0, 1]$. The value of n_m can be computed as follows:

$$N_m = \frac{\max(n_m)}{range(u_{i,j})} \times \left(\max(u_{i,j}) - U_{i,j}\right) = 0.25 \times \left(5 - U_{i,j}\right).$$

Note that in this case the greater the value of $u_{i,j}$ the smaller the value of n_m.

Maximum Value Dependency

A maximum value dependency $r''_{i,j,m}$ is a relation between an independent variable x and dependent variable y. Given that y is a variable belonging to a vector or a group Y, it is computed based on the following: $y = \max(Y')$, where Y' is Y without the value of variable y and $\max(Y')$ computes the maximum value found in the vector Y'. Note that this dependency is defined using a more generic format. In fact, we have to remember that in the previous chapter we have considered patterns that we called "special need." We have suggested that there are patterns that are directly applicable to a certain group of users, such as colorblind users. Such a pattern should be placed on top of relevancy list, as it is directly applicable to this particular user group. Therefore, maximum dependency is possible between all categories of variables.

Composite Dependency

In the case of composite dependency $r'''_{i,j,m}$ between an independent variable x and dependent variable y, we must note that it is composed of a set of relations s_d, where a particular sub relation Sub_n can belong to any type of relations. Moreover, Sub_n should be valid only for a subset of all possible values of x. In the case when a given Sub_n is found to be valid for all possible values of x, it should be promoted to a non-composite type of relation. For example, let us assume that the source can take any integer value in arrange of $[-1, 5]$, where -1 means unknown or undefined and rest of the values have a particular meaning. Moreover, the destination range is defined by $N_m \in [0, 1]$ where all values between 0 and 1 have a particular meaning. Considering a direct relation rule $r_{i,j,m}$, it does not make sense to attempt to compute the rule if the value of source is -1. Because the destination does not have a value representing an undefined condition, it is impossible to perform the computation of the rule. Therefore, we have now established that there is a single increasing dependency valid for source values $[0, 5]$ and a disabling condition valid for source value -1[7].

7.2.2.2 Computation algorithm

Although we have defined rules and some simple examples of their usage, the overall picture was never considered. More precisely, the core of the pattern selection is based on ideas gathered from recommender system. A suggestion made by the recommender system is based on a computation of the confidence of the result: the score. That is, if a user is described by values i and j, the confidence of suggestion of a result k is a sum of all confidences of rules that have i and/or j as a condition part and k as a result part. Considering the results obtained by C.Kim and J.Kim [2003], we propose to use sums of confidences of rules in order to allow for an item with more relations to have more weight. When comparing two common methods of applying association techniques, C.Kim and J.Kim found that AR+, method of sums of confidences of all available rules produces a better result than AR-MAX, method of maximum confidence rule.

[7] As we note here, a disabling condition is not considered as a rule. It is simply a manner to notify everyone that this particular value is undefined not by a mistake but rather by a choice of the specialist collecting and entering the data.

The input of the scoring process is a user matrix U for a given persona, where $U_{i,j}$ is a value of a $u_{i,j}$ user variable. We then propose to establish relationships between user variables $u_{i,j}$ and usability needs n_m. Therefore, given a relationship $r_{i,j,m}$, between a user variable $u_{i,j}$ and a usability need n_m, we can establish the confidence weight of n_m. More precisely, let us consider linear increasing and linear decreasing dependencies. For a linear increasing dependency $r^{|}_{i,j,m}$, the confidence for a need n_m is calculated as follows:

$$N'_m = \frac{U_{i,j} - \min(u_{i,j})}{range(u_{i,j})} + N_m .$$

In case of linear decreasing dependency $r^{||}_{i,j,m}$,

$$N'_m = \frac{|\max(u_{i,j}) - U_{i,j}|}{range(u_{i,j})} + N_m ,$$

where N_m is the original value of the need n_m, N'_m is the value of n_m resulting from evaluation of the relationship $r_{i,j,m}$, $\min(\ )$, $\max(\ )$ are the functions calculating the maximal and the minimal possible values for a given variable and $range(u_{i,j})$ is an absolute value of the difference between the maximum and the minimum possible values for a given variable $u_{i,j}$. Each dependency between user variables and usability needs must be considered. However, the order of the execution is of no importance. Once all the relationships are considered, we obtain a vector $\vec{n}$ that contains confidences for all usability needs.

Similar to the above, given $\vec{n}$ we attribute a confidence factor to a pattern characteristic $c_{i,j}$. That is, if a pattern characteristic $c_{i,j}$ is related with a need n_m that has a confidence value N_m then a pattern characteristic confidence $C'_{i,j} = C_{i,j} + N_m$, where $C_{i,j}$ is computed based on previously considered relationships. Recall that in chapter 5.3 we have considered the nature of relationships between usability needs and pattern criteria. We have found that the designers cannot identify the type of dependency existing between the two variables. However, our goal is to evaluate the importance and/or relevancy of a given pattern for the given persona in the current context. In lack of more detailed knowledge, we suggest to add the scores of the related needs to obtain the score of the pattern criteria.

Now, we consider a set of patterns described by a matrix P, where $P_{i,j}$ is a value of variable j in a pattern p_i. Here again we apply a similar process and compute the weights based on a weight of associated characteristic. The patterns are ranked based on their relevance p_c and a suggestion is presented to the user.

We must note that we have taken a set of approximations in the rule definitions and computation algorithm that have not been detailed by experts. However, we feel that currently proposed process is a good starting step towards a refinement of the scoring schema. Similar to neural network, farther research and analysis can allow for better approximation of dependencies, fine-tuning of individual values and computation algorithm, thus improving overall results.

7.2.3 Selecting Patterns Based on Personas

In the previous section, we have presented some of the details of the Patter Selection process. Let us now consider where these calculations and definitions will fit.

Overall, Selecting patterns based on persona consists of identifying a set of applicable patterns for a given design project per persona. That is, each persona is associated with its own list. Moreover, the task consists of selecting patterns from one or more libraries applicable of a given project. The steps are:

1. For a given persona, select a set of the most influential variables (and their associated values) for your design.
2. Compute the scores of the patterns following the proposed technique.

   - *Note that a labeled graph is a good representative solution that will help in the computations.*
   - *The start nodes should be user variables each containing appropriate value.*
   - *Edges should carry the values computed using corresponding dependency. Second level of nodes contains usability need scores and third level contains the pattern criteria scores.*
   - *An Excel table or any other similar tool can be used to compute the pattern relevance scores and order them accordingly.*

   2.1. For each usability need, compute the resulting score by summing up the scores from all associated dependencies.

   2.2. For each pattern criteria, compute the resulting score by summing up the scores from all associated dependencies.

2.3. For each pattern, compute the resulting score by summing up the scores from the two associated pattern criteria.

   – *Recall that there are two pattern criteria, where the second one is considered secondary and only "half" of the effect of the first. Thus, the resulting score is a sum of the primary criteria and half of the secondary one.*

2.4. For each pattern containing a special need variable associated with respective user variable, compute the score following *max( )* equation defined earlier.

   – *Recall that some users may belong to a special needs group (like color-blind) and there may be some patterns specifically designed for these users.*

3. Organize the patterns in a table ordering them in descending order where the patterns with the highest scores come first. Note that ordering method chosen for patterns with the same scores is irrelevant as all of them have the same relevance to the given persona.
4. Repeat the above steps for each persona.

Let us now return to our case study in order to gain hands-on experience with the process. As part of the new provincial initiative, the city of Montreal wants to increase accessibility to information and services for public. It has offered financial support for various initiatives targeted at populations with special needs. It so happens that our *crazy shopper* project has evolved on its way and management is ready to take the challenge considering substantial financial support provided. As a result, the specialist found that the population has somewhat changed now and we need to design for another persona:

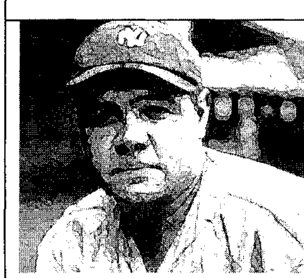
Benoit Lalonde, a 40-year-old man with low literacy. Given very particular needs of Benoit, we decide to design a game specifically dedicated for this group of population.

Case Study

Following is an extract from description for Benoit. In variables, we note that Benoit has a low education level and literacy.

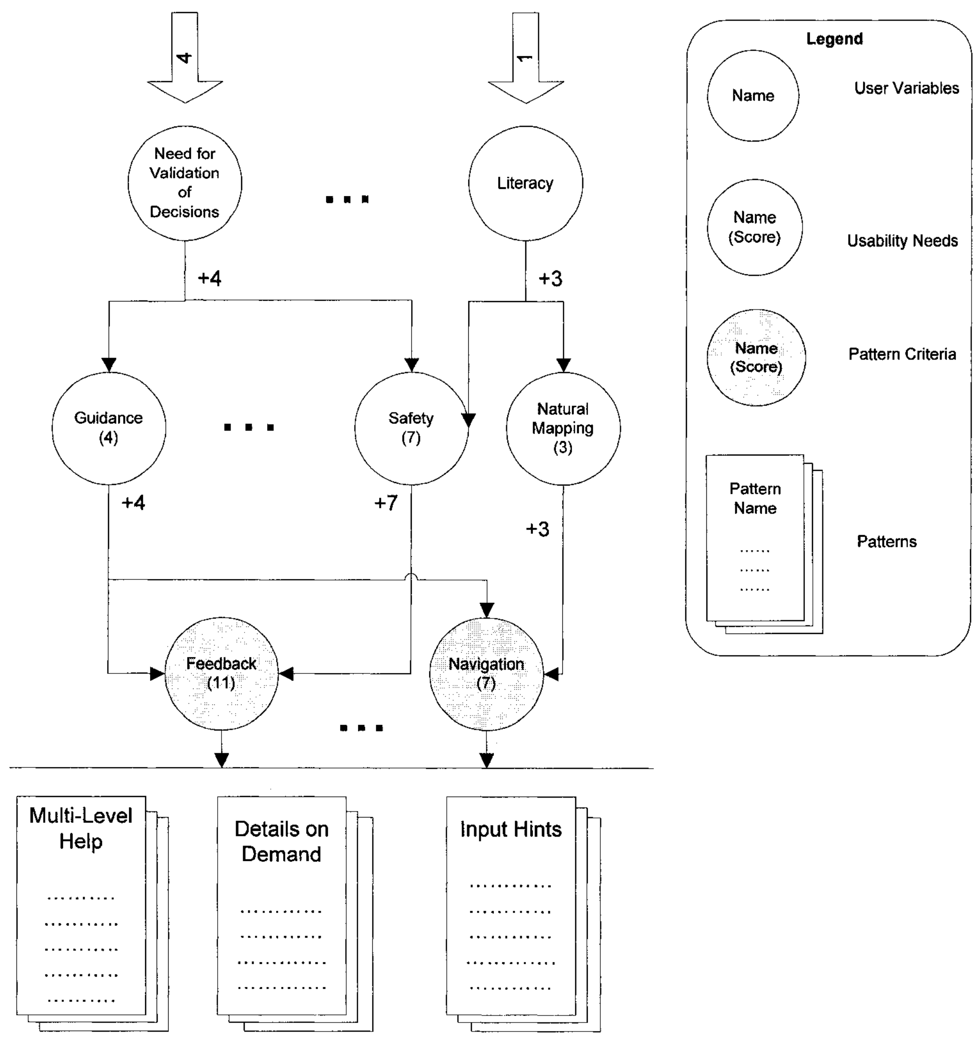
User Variable	Value
Literacy	Low (1)
Validation of decisions	High (4)
Education Level	Elementary (1)
Linguistic Ability	Average (2)
Computer Experience	Basic (1)
Guidance	Somewhat high (3)
Learning Speed	Somewhat low (1)

As a result, he belongs to a special needs group with low literacy. Following is a part of associated textual description.

<table>
<tr><th colspan="3">Benoit Lalonde</th></tr>
<tr><td rowspan="2"></td><td colspan="2">I want to be able to spend some time with my children playing games they like to play and speaking about things that interest them.</td></tr>
<tr><td>General Profile</td><td>Benoit is a 40 years old worker at Bombardier. For his entire life he has been working for this company. He lives with his wife and 3 children. He has a difficulty writing and understanding most forms and documents. […]</td></tr>
<tr><td>Goals</td><td colspan="2">[…] Increase his savings and spend some time with his children.</td></tr>
<tr><td rowspan="4">Scenario</td><td colspan="2">[…]</td></tr>
<tr><td>Spec. needs</td><td>[…]</td></tr>
<tr><td>Features</td><td>[…]</td></tr>
<tr><td>Int. details</td><td>[…]</td></tr>
<tr><td>Demographics</td><td colspan="2">[…]</td></tr>
<tr><td>Knowledge and Experience</td><td colspan="2">Benoit is a native French speaker. He has learned spoken English during the years at work. […]</td></tr>
<tr><td>Psych. profile and needs</td><td colspan="2">[…]</td></tr>
<tr><td>Attitude and Motivation</td><td colspan="2">He does nor like computers but accepts the need to learn some things for work and to be able to communicate with his children</td></tr>
<tr><td>Special Needs</td><td colspan="2">He has no disabilities but belongs to a special user group, low literacy users.</td></tr>
</table>

Case Study

Now, we are ready to go through Pattern Selection sub-process. Based on Benoit's interaction behaviour, context of user, and needs, we select two influential variables *literacy level* and *need for validation of decisions*. Following the above proposed process, we need to compute the scores of usability needs, then pattern criteria and then patterns in the available pattern library.



For example, Benoit has a literacy level of one. There exists a dependency between *literacy level* and *natural mapping*: "when literacy level is low (1) natural mapping, safety [...] increases by 3."

Case Study

Therefore in our example above, we can compute the score of natural mapping: it is 3 as it has only one dependency that we just evaluated. On the other hand, score of *safety* is a sum of scores from a set of dependencies. In our case, we add the dependencies from both user variables, thus giving us a score of 7.

Once we have computed all the usability need scores, we can compute the scores of the pattern criteria. *Navigation* has dependencies with *guidance* and *natural mapping*. Therefore, the resulting score of 7 is a sum of the two usability need scores.

Now, we compute the scores for each pattern in the library. For example, *details on demand* pattern has as a secondary criteria as feedback. Therefore, its relevance score will be half of this value and equal to 5.5. On the other hand, *multi-level help* pattern has the same criteria as primary and its score will be 11.

Finally, we should consider patterns, if any, which address the needs of the special group that Benoit belongs to. These patterns will be placed at the top of the pattern list.

By computing the scores of all the patterns that we have selected from multiple libraries [Tidwell 2002; Javahery and Seffah 2002; Welie 2003; Wilkins 2003] and have created ourselves, we obtain an ordered list of suggestions (see Table 19).

Table 19: An ordered subset of suggested patterns

# (in library)	Title	Short description
38	Simple text	Organize content using simple and short text.
39	Alternative orthography	Write content phonetically (phonetic spelling)
40	Audio communication	Provide content in audio format
19	Filter	Provide filtering facilities to reduce number of items

# (in library)	Title	Short description
31	Card Stack	Use surfaces stacked together to group content
5	Multi Level Help	Provide help techniques to support variable needs
17	Reference context	Refer graphical objects visually to reference context
29	Wizard	Guide user through the tasks and maintain visibility
18	Details on demand	Use a separate window to display item details
7	Input Hints	Place an example beside data input field
14	Contrasting Font weights	Separate levels of info by varying font weights
11	Smart Selection	Automatically select coherent groups of items
1	Tree Table	Combine a tree and a table to illustrate hierarchical structure of data
2	Overview + details	Place an overview next to a zoomed view

Let us now analyze the results obtained in order to understand the process better. The top three selected patterns are all patterns for a special needs group with low literacy. They are all focused on providing alternative ways of acquiring information either by simplifying the content or by using voice as delivery means. Then we see a set of patterns mostly containing solutions to complexity or support problems. Filter and Details on Demand can be both used in order to simplify the provided content; thus, facilitating access to information. Moreover, Multi Level Help, Wizard and Input Hints can all be used to support the user and provide the so needed reassurance for Benoit.

7.2.4 Modifying Selection Parameters

In the previous section, we have described the main scenario for pattern selection and have experimented with it using our case study. It is possible that the specialist during the process will gain a deeper knowledge of the persona and realize that some of the parameters used in Pattern selection need to be modified. Pattern libraries, influential variables are only two of the possible modification areas. The decision here should be guided by the insight gained during the first Pattern Selection iteration. A particular care should be exercised when modifying the list of influential variables. As some variables, may favorable compliment the overall description of a persona but may not be representative of the corresponding user group. Such age (40 years old in our case study) may be used to justify users goals (spend time with children), but should not be used as influential variable if it does not reflect the age group of the represented population.

Case Study

For example, in our case study we have stopped at a point when we have obtained an ordered list of suggested patterns. Although we have obtained a plausible list, we note that this persona the same as the entire group requires *somewhat high levels of guidance*. In fact, we can argue that there is a hidden dependency between guidance needs and literacy level. It is possible for a user to feel unsafe and require great amount of guidance in a hard-to-understand environment specifically because of low literacy levels. Therefore, we modify influential variables set by adding *guidance*.

When we reiterate through Pattern Selection process, we find that the list did not change much (see Table 20).

Table 20: Ordered subset of suggested patterns (iteration 2)

# (in library)	Title	Short description
38	Simple text	Organize content using simple and short text.
39	Alternative orthography	Write content phonetically (phonetic spelling)
40	Audio communication	Provide content in audio format
5	Multi Level Help	Provide help techniques to support variable needs
19	Filter	Provide filtering facilities to reduce number of items
31	Card Stack	Use surfaces stacked together to group content
18	Details on demand	Use a separate window to display item details

# (in library)	Title	Short description
29	Wizard	Guide user through the tasks and maintain visibility
7	Input Hints	Place an example beside data input field
11	Smart Selection	Automatically select coherent groups of items
14	Contrasting Font weights	Separate levels of info by varying font weights
17	Reference context	Refer graphical objects visually to reference context
1	Tree Table	Combine a tree and a table to illustrate hierarchical structure of data
2	Overview + details	Place an overview next to a zoomed view

However, some patterns have moved higher: *Multi-level help*, *Details on Demand*, *Wizard* etc. are now two or more position higher. In general, we see that by adding *guidance* to the Pattern Selection process we have put a greater focus over it. Therefore, now there is a grater chance that the patterns that will be selected during conceptual design will contain a greater amount of patterns dealing with guidance.

7.2.5 Filtering: Domain and Task Based Filtering

In the previous section, we have described Pattern Selection process and scoring technique as part of it. We have assumed that we have a large pattern library most probably aggregated from various sources and domains of application. Then we have performed a set of computational and analysis steps that resulted in an ordered list of suggested patterns. However, we have not considered two most important factors of the design context: domain of application and tasks executed by our users.

Filtering Patterns step consists of reducing pattern set based on a priori knowledge about domain and context of design. It is possible that suggested pattern list contains more than 80 patterns that may contain in various proportions Web, Gui, Mobile, 3D Visualization and other domains of application that are not all necessarily useful for the current project. Moreover, the application at hand may not be suitable for certain interaction models that will eliminate some of the patterns.

Fro example, in our crazy shopper example, we are designing a mobile phone application. Therefore, it should be lightweight and easily manipulated on a simple cell phone keypad. Therefore, we can safely omit all patterns that are exclusively focused on 3D Visualization applications. Moreover, we do not expect any multi step installation or configuration processes; therefore, *Wizard* pattern can be eliminated. Similarly, Tree Table is a complex GUI and Web pattern requiring large screen size and used to demonstrate hierarchical relationships among multidimensional data items that can be eliminated.

Overall, we suggest considering large groups of patterns first. Similar to our case study, we should first eliminate patterns that belong to certain domain (if applicable). Only then, when the list is much smaller, a designer should consider the patterns from top to bottom in the list and eliminate all candidates that clearly do not apply for the given project. We must note that a designer should not consider the relative position of a particular pattern as a reason for elimination. A list of patterns is only a suggestion guiding the designer. It is possible that low score patterns will still be applicable for the given group of users and will enhance overall design. In fact, the scoring system orders patterns based on some importance or relevance assumptions where higher score means more relevance but does not suggest that low score patterns are not applicable.

7.3 Pattern Composition

This phase consists of a design activity represented by one step: Composing patterns. Recall that we have already discussed in chapters 2 and **Error! Reference source not found.** some of the available pattern libraries and techniques for pattern composition. While we do not necessarily restrict the designer's choice of methods used at this stage, we have already built the majority of prerequisites for application of Pattern-Oriented design (POD) model.

The proposed model has been described in various publications [Javahery and Seffah 2002; Javahery et al. 2006; Javahery 2007(a); Javahery 2007(b)]. POD model helps designers to break down the conceptual design into multiple levels of concern. For the website design, four steps have been defined:

1. Define the structure of the site with architectural patterns
2. Define overall structure of each page with page manager patterns
3. Define content elements with information containers
4. Organize the navigation with navigation patterns

During all stages of the design, the specialist should exploit the relationships that we have defined earlier. Recall that we have defined a pattern variable called *relationships* identifying how a given pattern associates with others. We have also defined five relationships: Similar, Competitor, Super-ordinate, Sub-ordinate and Neighbouring. For example, the designer can use the Sub-ordinate and Super-ordinate relationship to navigate from one step to another in the POD model. At the same time, Similar, Neighbouring and Competitor can be used to make design decisions within a particular level of the POD model.

Similar to previously described Persona Creation and Pattern Selection, Pattern Composition is an iterative process. At this stage, the amount of iterations depends on the designer's resources and experience. However, from the first iteration the designer will be able to produce a comprehensive design using patterns as building blocks. This design then can be farther detailed by attaching it to a particular application and context domain.

7.4 Summary

In this chapter, we have described and explored the proposed P2P process. We have described three major phases: Persona Creation, Pattern Selection and Pattern Composition. Each phase resulted in an output that was used as input for the next phase.

Persona Creation phase required initial context, demographic and other user information to be accumulated beforehand. It consisted of grouping users based on their similarities when considering a small group of influential variables. This step resulted in a set of refined persona representing a particular user group.

Pattern Selection phase required final set of persona and a pattern library to be present. It consisted of a scoring technique that calculated a relevance score for each pattern. The output of this step was a set of patterns ordered based on their importance for the given persona and filtered based on the domain and task knowledge.

In the Pattern Composition phase, we have concentrated in using patterns as building blocks to produce a conceptual design. This step required the pattern list obtained from previous step and some expert knowledge about POD. It resulted in a conceptual design that can be further detailed based on application domain and context information.

We have explored the process through a perspective of a simple case study that allowed us to experiment with most parts of the process. While we have not concentrated on our perceptions during the process description, we have found that some steps required a greater involvement from the designer side than it was initially expected. For example, computation of scores required a large amount of time and consternation on the behalf of the designer. Therefore, as we have already suggested in sections 3.1.2 and 5.3, there is a strong need to consider a support environment that will further alleviate the load on the designer. In the next chapter, we will explore a prototype environment called P2PMapper tool designed to support the specialist during the entire design process.

8. P2PMapper Tool

As we have mentioned previously, the proposed process involves a set of repetitive and time-consuming tasks. Therefore, we have attempted to build an initial prototype that will support the process and alleviate the charge placed on the HCI specialist during the design phase. In general, the tool provides the designer with three major functionalities: user data entry system, clustering capabilities and pattern selection (see Figure 11).

8.1 Architecture

User data entry system is comprised of a set of widgets and access points that allow for easy user entry, viewing and modification of information. Similar to the described process, the information can be of two formats: predefined set of discrete value characteristics and narrative text. Moreover, all the information is stored in XML format that can be edited by other tools and converted into a variety of markup languages. At the end of this step, the specialist has a user base that can be manipulated by the tool.

Once the data is entered, the tool provides the designer with clustering capabilities. P2PMapper tool allows the designer to automatically group users based on two different algorithms: K-Means clustering and Sequential clustering. At the same time, the user is provided with an interactive interface that allows modifying newly formed clusters on fly. Moreover, all available methods have been designed in order to facilitate user grouping and allow for multiple iterations over a short span of time. For example, quantified persona set is automatically created as soon as the designer starts clustering. It is also automatically modified when clusters are adjusted. However, textual description and persona extension process requires designers' involvement and creativity. Therefore, the result may be an extended persona if the process is followed or simply quantified persona set to be used for pattern selection.

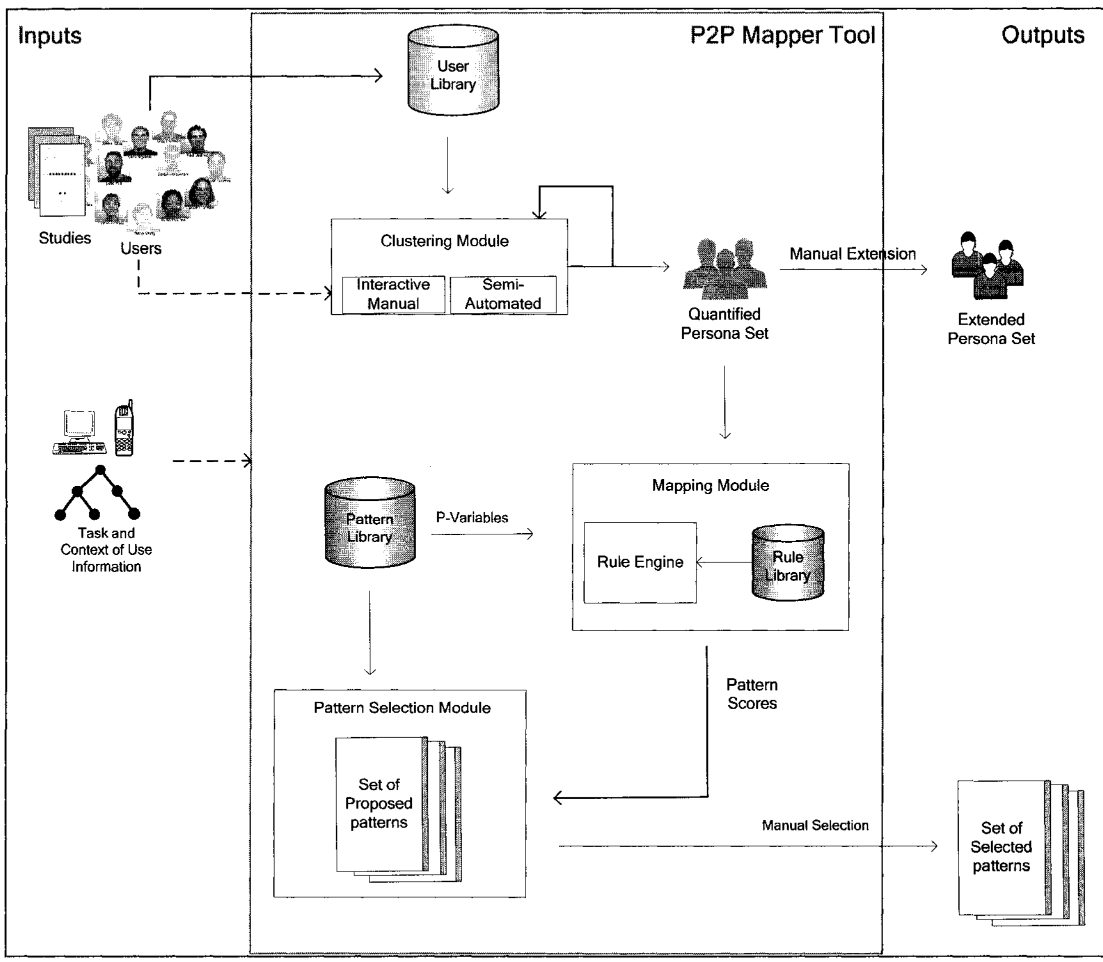


Figure 11: Overview of P2PMapper tool

Once quantified persona set is created, the designer can use mapping module in order to search for a set of suggested patterns. This step is fully automated and requires only for initial configuration. The tool, based on selected values of each persona, generates a set of patterns ordered by their importance. The mapping module operates by applying a large set of rules (based on expert knowledge), that compute the score of each pattern for the given persona. The result is in an ordered set of patterns with the highest score pattern coming first. Finally, the designer can filter and reorganize the set based on some general criteria, such as domain, pattern type, source library etc.

8.2 Tool Support for P2P Process - an overview

As we can see from the architecture, the tool is build around the process. It is centered over a set of key activities: persona creation and pattern selection. However, each phase slightly differs from its equivalent in the process and Pattern Composition phase in not implemented (see Figure 12).

Let us consider Persona creation phase. In the process description, we have assumed that all user information has been already compiled, formatted and is available at fingertips of the designer. However, it is sometimes hard to compile and present the information for multiple users in exactly the same format. Therefore, we have added an extra step to the persona creation: Data Entry. This step consists of dissecting and extracting the information required for sufficient user description based on the design problem.

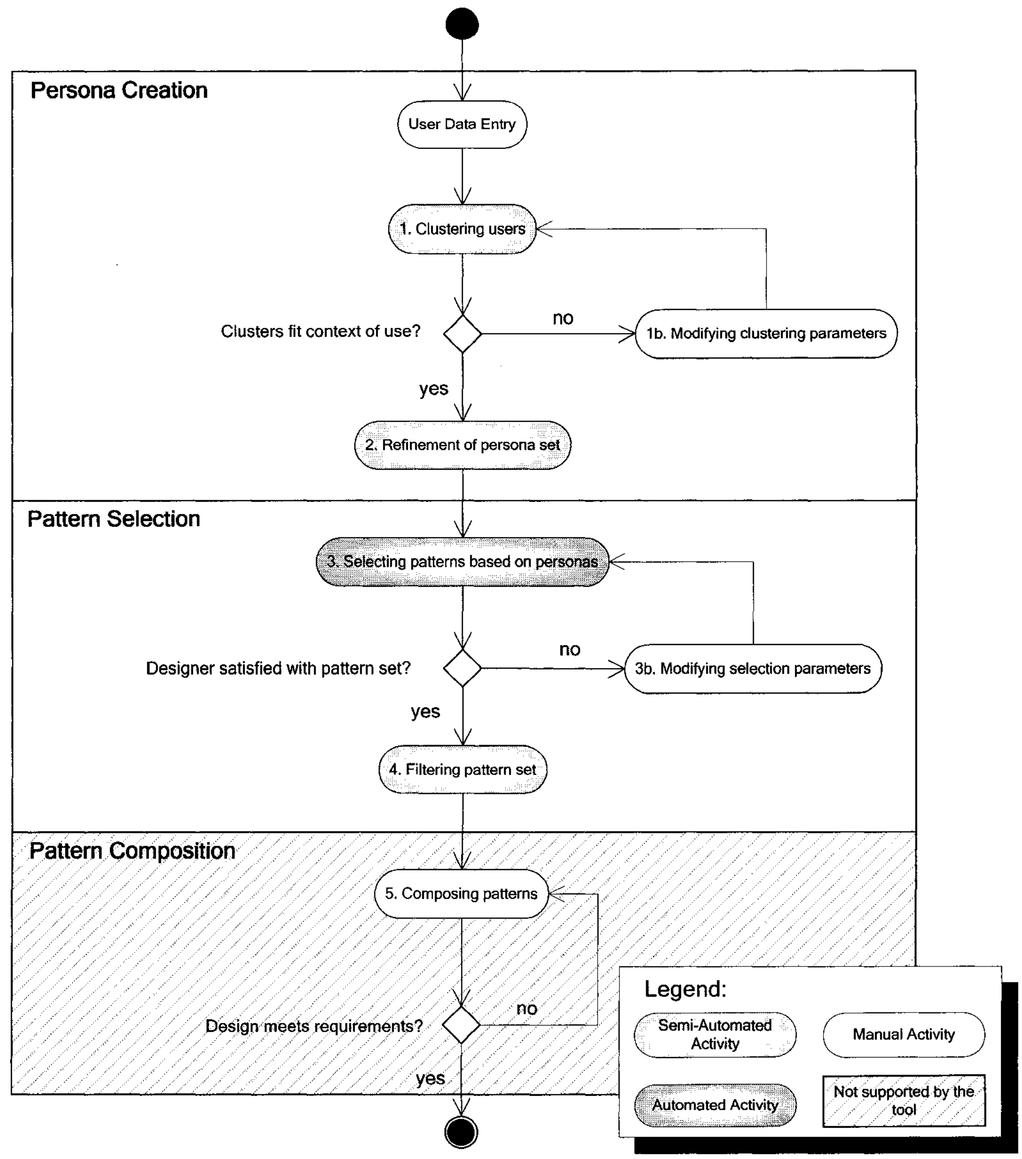


Figure 12: Key activities of tool supported P2P Process

In contrast with persona creation, pattern selection phase is the most tedious work that requires little creativity. Therefore, we have attempted to automate it: a single click of a button can produce a set of suggested patterns. In fact, the tool allows for rapid iterations and eliminates the necessity for manual calculations. However, the designer is provided with a trace of decisions made in order to make an enlightened decision on modification of mapping parameters (if required).

Finally, pattern composition phase is a highly creative process based on previously acquired design experience. As such, it cannot be automated given the current state of the art. While it is possible to foresee a design environment specifically implemented for this purpose, in order to develop such a system great resources should be allocated to process discovery and usability prototype testing. Moreover, it is yet unclear whether such systems will help or on the contrary slow down the designer. At the same time, pen, paper and a large board seem to be a good and simple alternative that has been used for this purpose for years. Therefore, in the scope of the current research we have decided to offer no tool support for pattern composition phase.

In summary, we have described an overall functionality of the tool and have overviewed the key phases of the tool supported P2P process. We have designed and implemented a set of strategies in order to achieve maximum flexibility and provide the designer with all the tools necessary to apply the process. The tool has been designed with changes in mind and with a goal to favor creativity while alleviating the load by automating tedious and repetitive tasks. This has been done through a set of strategies such as customizable views, interactive user clustering process and automated pattern selection.

8.3 Views in P2PMapper Tool

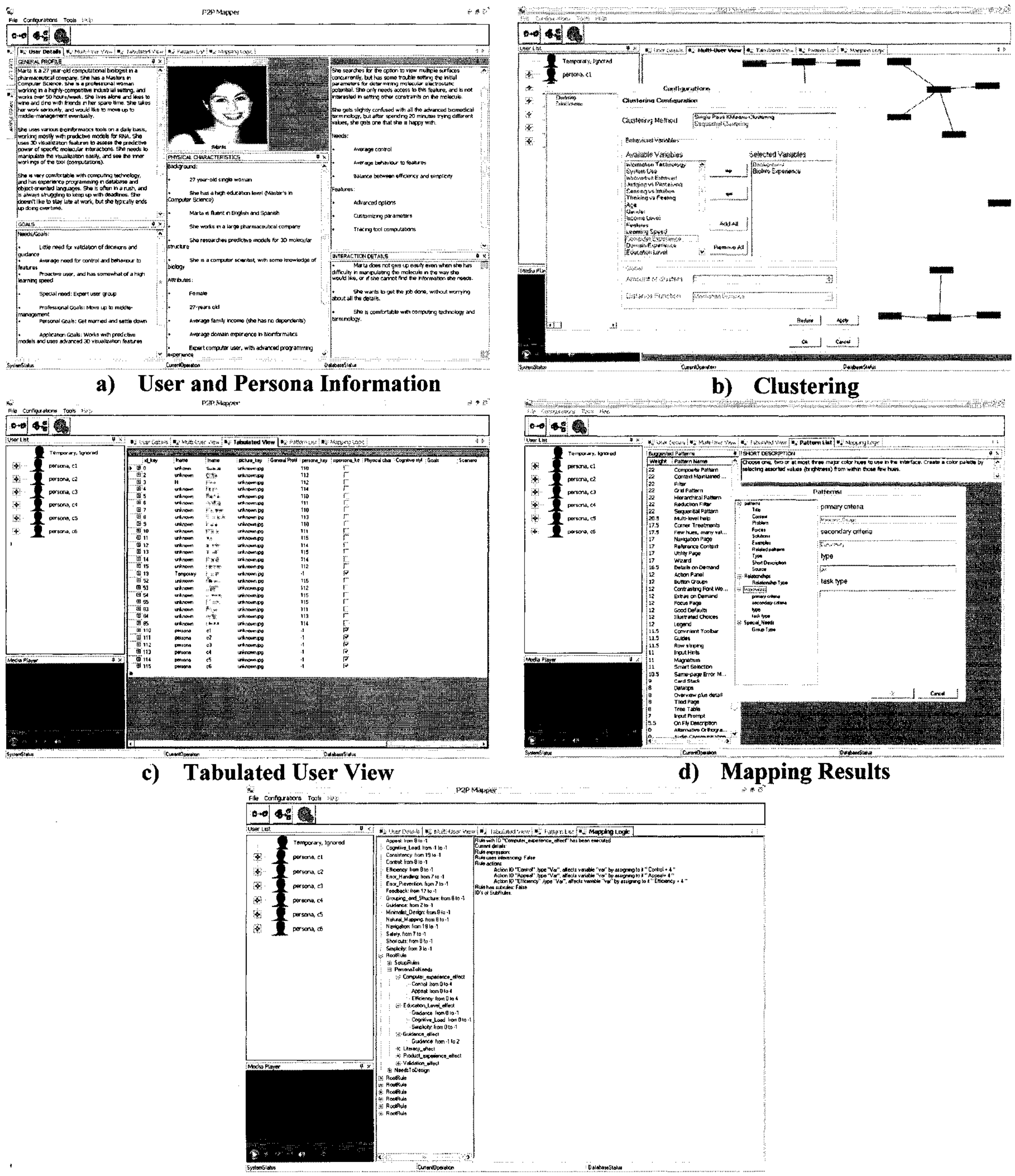
The tool is organized around the key activities of the user as described above. It contains five views, where three are used for key activities and other two provide complementary and alternative information. The first view (see Figure 13a) is used for data entry and review of persona and user information. It presents a customizable view of all the textual information and allows for easy access to user variables.

On the second view (Figure 13b), the specialist has access to clustering results represented using color and special encoding method: the users are grouped around a persona object and have a similar color. In this view, the designer has an option to use drag and drop to move the users from one cluster to another.

In the fourth view (Figure 13d), the designer has access to clustering results and pattern information. This view presents an ordered pattern list for each persona and additional information on a particular pattern the designer decided to view in detail.

The third and fifth views are complementary (Figure 13c and Figure 13e). The third view presents user and persona information in tabular form. It is mostly intended for an expert use specifically in cases when some modifications to user format have been made and need to be verified. On the other hand, fifth view is a basic trace of the Pattern Selection logic. It presents all hierarchically organized operations in the same order as they where executed by the tool.

Overall, most of the views can be customized and reorganized. In fact, the tool uses floating windows interaction paradigm as it has been used in Visual Studio 2005 and many other software applications. In addition, user and pattern formats and all supporting GUI elements are generated on fly from the corresponding xml file. Thus, this allows for future evolution of pattern and user/persona descriptions without a need to change the tool. Let us now consider in detail the key activities supported by the tool.



a) **User and Persona Information**

b) **Clustering**

c) **Tabulated User View**

d) **Mapping Results**

e) **Mapping Logic**

Figure 13: Combined view of P2PMapper tool

8.4 Grouping users: interactive and semi-automated process

One of the key points of the proposed process is clustering. Following the process presented in section 7, clustering can be described as a process consisting of an iterative grouping of users based on some parameters and adjustment of these parameters based on quality of results obtained. In order to alleviate the load put on the designer during user grouping phase, we have designed and implemented a set of automated grouping techniques. Additionally, we are conscious of the fact that automatic processes cannot account for all possible exceptions; therefore, we have provided means for manual clustering and a possibility for interaction with automatic clustering algorithms. The designer can use automatic clustering methods to produce rough results and gain some insight into user groups. At the same time, a process can be followed exactly and semi-automated clustering can be performed by using per variable (or as we call it sequential) grouping and manual drag and drop operations.

8.4.1 Automated clustering methods

As a means to automate this step, we can apply dynamic and fuzzy clustering techniques [Suryavanshi et al. 2005] on the set of user variables defined in our user model, to arrive at the number of different personas that need to be created. For the purpose of our analysis, we can limit clustering techniques to two groups: hardbound and overlapping.

Hardbound clustering techniques consist of constructing a set of distinct groups, called clusters, with a goal to obtain a homogeneous set of groups. Therefore, hardbound clusters assume that we can clearly draw a bound between groups and that no data items can be in more than one group. Such techniques are most often lightweight. On the other hand, overlapping clustering techniques essentially assume that all data belongs in part to all clusters. The major assumption is that there is no real bound between data sets. Overlapping clustering usually provides the most nuanced results and the best fits with a cost of large resource usage. For example, fuzzy c-means clustering technique is similar to k-means but provides additional information, such as "how close each data item is to all constructed clusters." However, the process that we are attempting to automate takes no advantage of the additional information obtained from fuzzy clustering techniques. Therefore, both solutions seem to be applicable at this point.

On the other hand, when we consider automated grouping solutions we should also consider their tolerance to noise. In general, by noise we can consider any data item located relatively far from any other data item forming a cluster. In field studies, it is possible that due to external constraints, like sickness, some users will have an abnormal behaviour and performance, will fall out of their cluster, and should be eliminated. In general, this task is performed by a designer that analyzes the data and attempts to obtain information explaining the behaviour of the user. In a case of automated clustering, some extended methods, like extended fuzzy c-means (eFCM) allow for detection and elimination of noise.

However, elimination of noise may result in elimination of legitimate users. It is common for HCI to have small data samples. It is also possible that the distribution of users, although random, is not always equal amongst different groups and categories. Therefore, it is possible to have a very small number of users, distant from any other large cluster. In this case, most common clustering techniques with low sensitivity to noise will ignore these users. Therefore, methodologies designed for large sets of data with low sensitivity to noise seem to be unsuitable for the purpose of the project.

K-Means Clustering

As a result of the above reasoning, we have concluded that k-means is the most appropriate technique at a current stage of development because it provides all the required resources and is implemented in variety of languages as public source.

Given the technique we have selected, the designer is invited to define a set of user variables that will be considered for grouping. In addition, the designer must decide on how many clusters there should be. In fact, K-Means clustering technique requires a priori knowledge of the amount of clusters. Although, the designer may not know how much groups should emerge it is possible to predict a range of solutions. Considering the fact that clustering can be performed with a click of a button, the designer may try possible arrangements for each solution. For example, the designer may predict that in current context there should be from 3 to 7 personas. Therefore, the designer may attempt clustering in 3, 4, 5, 6 and 7 groups. By observing the results obtained, the designer will be able to see which users always are grouped together. Moreover, the designer will be able to see which groups seem to be most homogeneous for the smallest amount of groups possible. Based on this information, the designer will be able to select the amount of clusters that resulted in the best groupings.

Sequential Clustering

Although automated clustering technique with unknown reasoning behind may help the designer to gain invaluable knowledge on the affinity of users and their relative distance based on their values, we found that automatic clustering is not sufficient to accommodate all envisaged cases. More precisely, grouping users by their relative distance sometime may result in a user falling in a group that he/she does not belong. For example, a user may be found to be an equal distance from two existing groups based on an overall distance calculated for all variables. This user will them be placed in one of the groups based on the random choice or first come first serve basis. Given the above possibility, the designer needs to be able to verify and understand possible decisions the algorithm has made to classify the users. However, for large user groups or large variable sets (more than 3 variables with 5-7 possible values each) the quantity of information to absorb and analyze is tremendous and hard to grasp.

Therefore, we propose an alternative clustering method that mimics designer's strategy to build personas. This method breaks down a user group into a set based on the possible values of the first variable. Then each obtained set is broken down based on possible values of the second variable. The last step is repeated for all variables; thus, resulting in a large set of clusters each containing a unique combination of values for a selected set of variables. Given the fact that this method considers only the values present in the user data, the resulting set of clusters can be quite small if user data is condensed around a small range of values. Similar to the process described in the previous chapter, the designer is responsible for analyzing clusters obtained and combining, removing or modifying them.

Recall, that we have already considered such example in the previous chapter when we discussed our *crazy shopper* application and the persona creation process. In our case study, the designer was required to select set of variables and group the users based on the variables. Then some groups were eliminated mostly due to redundancy. Finally, some new variables were added and the process has been repeated. The tool allows for the execution of the same process without any major modification. In fact, the only difference is that there is no need to consider the range of values for a given population during clustering: the tool will not produce empty sets for values that are not represented by real users.

8.4.2 Facilitating manual persona creation

When the specialist is required to perform a manual adjustment to the proposed grouping or perform the entire grouping manually, the tool employs a drag and drop interactive paradigm to facilitate user manipulation.

We have reviewed a large set of models, computer or paper based, facilitating grouping of a set of complex entities. [Tidwell 2005] argues that color, shape and relative spacing can be used to produce a feeling of similarity between items. At the same time, [Nardi et al. 2002] have used a model where friends of a given user were distinguished by color and relative distance based on their importance or belonging to a particular subgroup, such as coworkers. In our case, we have adopted the same strategy (see Figure 14).

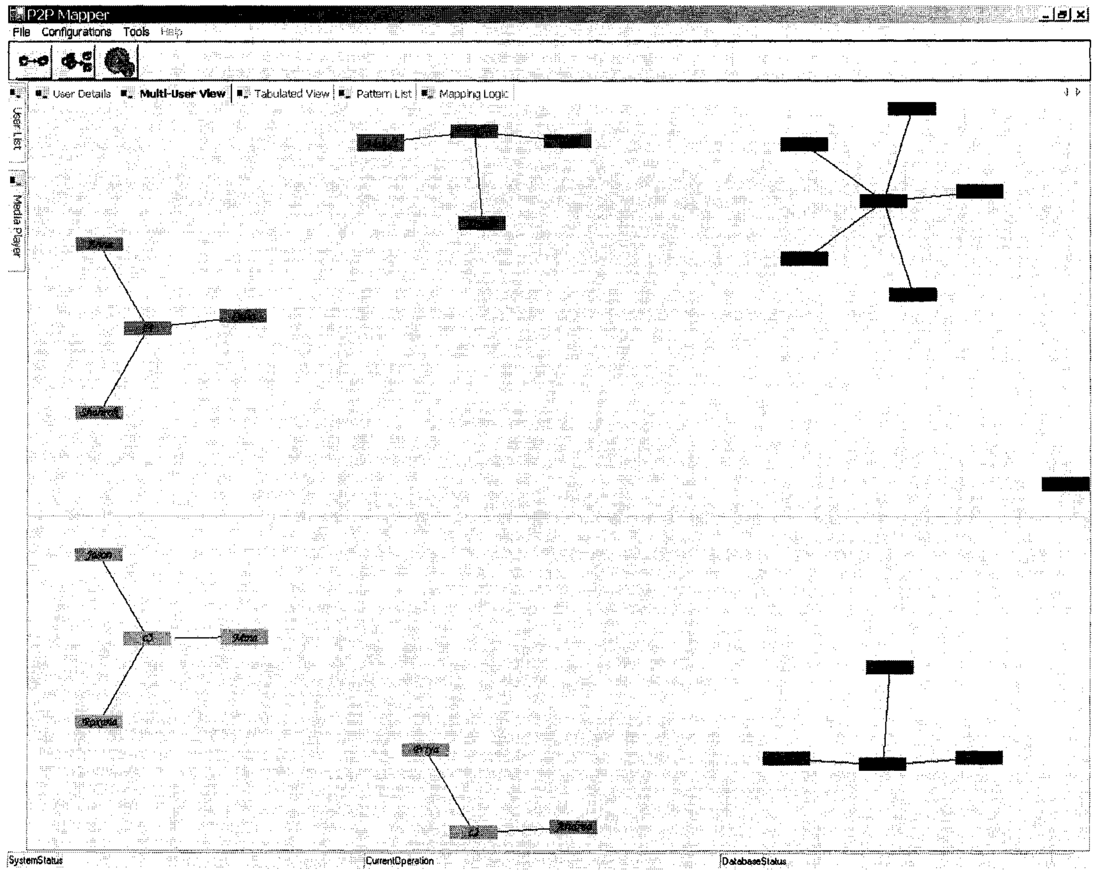


Figure 14: Multi-User View in P2PMapper Tool

More precisely, each user is represented by a name and an optional picture. Each user is placed closer to the users in the same group than to the users from the other group. A persona representing this group is placed at the center of the group. Moreover, the users can be optionally connected to their persona by a line. Finally, a border around each user is colored based on its current location. So, if a group is represented by a persona with a green border all the users in the same group will have a similar border color. When assigning a user to a particular group, the specialist can drag the user closer to this group, resulting in changes of border color and association line (optional).

Moreover, we note that there is a one empty user set located on the right. This set is called "ignored" and is used as a garbage can that stores all the users that should not be used in future clustering but are part of the user pool. Recall, that in our crazy shopper example we needed to eliminate some user groups because their behaviour was encapsulated in other user groups. In P2PMapper tool, this can be accomplished by dragging the users close to the "ignore", which results in this users being ignored by all the modules in the tool.

8.5 Pattern Selection

Following the process described in section 7.2, we have attempted to find a technique that will allow us to implement a scoring system with an ability to perform basic logical operations based on a pre-established set of rules. Many AI techniques, like Decision Trees, Neural Networks and Inference Engines, propose tools and techniques that can allow for implementation of a complex set of relationships. However, each technique is suited for a different purpose and application and has some advantages. For example, Neural Networks allow for training and adaptation based on the feedback received. Inference Engines allow to perform logic analysis and thinking based on a predefined set of rules.

In order to support designers in the learning process and provide sufficient information on the rationale, the method or the tool selected should allow for traceability. More precisely, the designer should be able to understand and see what logical steps lay behind a particular decision. This knowledge is crucial for appropriate application of the tool and process support.

More precisely, automation of scoring and pattern selection hides in back box all the rationale described in a process. While this strongly reduces the burden on the designer, it also hides the rationale and crucial knowledge. This may result in confusion and distrust towards results from automation. For example, a tool may produce a list of patterns ordered by their importance based on a large set of personas originating from a large initial set of user. Given the amount of information the designer should absorb and know, we can assume that he/she will be overwhelmed and will not get a clear picture of the entire design. Therefore, it is possible to feel that the system has produced partially erroneous set of patterns. In that case, the supporting environment should provide for means to justify every decision made and compensate lacking knowledge, such as human readable trace of logical steps that have lead to a particular decision.

At the same time, we should consider the supporting tool evolution. We have to use tools and methods that allow for future adjustment and modification. More precisely, in the process we have defined a format and an interaction style of rules. Additionally, we have established an initial set of rules. However, we envisaged possible evolution of the patterns, rules and user model as the HCI field matures and knowledge accumulates. Similarly, the supporting environment should allow for easy modification and evolution. Therefore, the tool used for support of selection should provide means for easy addition and modification of rules and overall reasoning.

The Decision Tree is one of the most popular classification algorithms in current use in Data Mining and Machine Learning. It is also commonly used in operations research (decision analysis). Although the definitions of decision trees differ, their usage is similar. Decision trees are used as a method to classify the information at hand based on a predefined model or graph structure represented by a tree. In general, the node is defined as a decision point that will result in a selection of a particular path defined by an edge. Decision trees are commonly used when a complex decision has to be made or/and certainty of an outcome calculated. They allow for modeling binary or discrete output. Decision trees need to be constructed before use. Based on the domain of application, training or manual tree building methods are used. From the first sight, decision trees seem to be a possible implementation of the process. However, the relations some times affect multiple variables. Therefore, modeling each relation will require extensive effort and restructuring with additional design constrains. However, this brings us to another method commonly used in artificial intelligence (AI): Neural Networks.

Neural networks come from an effort to closely model the functioning of a human brain. That is, a neural network can be visualized as a graph structure where each node is defined as a neuron that performs some action. The output of the network is a result of cooperation of all interconnected neurons. However, the unique feature of this method is that it continues to function when one neuron fails. More precisely, Neural Networks are error tolerant. Typically, Neural Networks are used to model or to discover complex relationships. A neural network is initially "trained" or fed large amounts of data. This allows each neuron to determine its behaviour based on the input received. In making determinations, neural networks use several principles, including gradient-based training, fuzzy logic, genetic algorithms, and Bayesian methods. Considering the fact that we are attempting to construct a module able to establish complex relationships between persona and patterns, Neural Networks method seems to be a good candidate. However, the major drawback of this method is requirement for extensive training. In our case, this means that for each type or format of a project it may be required to retrain the engine. The time and resources necessary to train a Neural Network may make the use of supporting environment more cumbersome than applying the process manually. As a result, Neural Networks seem to be a sound candidate if there are some means to train the algorithm for each project automatically, without designer's involvement.

After considering a few methods from AI field, we have turned to a world of business applications. One of the common methods used in complex business applications to automate decision process is Inference Engine. Inference engine is rule based reasoning system that executes a set of rules given a current content data store. It performs three basic actions: match rules that apply, select rules based on hard-coded or run time determined strategy, and execute selected rules. Usually actions determined in rules affect data store values but may also generate new events or trigger more rules. As we can see, rule engine relies on two information blocks: data store and predefined rule set. Data store defines all objects that rules can access and manipulate. In simple form, rules are defined in a format of condition and action. For example, a simple system to determine the state of the road based on the weather condition may have two objects defined: weather condition and road condition. The rule is formulated as follows: if weather=rain then road=wet. This simple rule engine will assign "wet" to the road conditions when the weather is "rain." Overall, rule engine seems to be good candidate for the implementation of the pattern mapping. In fact, related process artefacts have been defined in format of rules that affect some objects (variables). From the three methods analyzed above, Inference Engine seems to be most suitable for supporting Pattern Selection sub-process.

Therefore, we have converted all previously defined rules in a format suitable for an inference engine. The resulting module computes the relevance scores of all the patterns currently available in the tool library and produces a sorted list. As it was defined in a process, the tool computes the scores of patterns per persona. Furthermore, the tool produces a trace of decisions and rules that have been executed that allows a designer to verify the proposed solution.

It is important to note that one major advantage that the tool confers to the designer is the access to information at his/her fingertips. Recall that we have assessed (in section 3.1) experience dependency and the need to search for solutions as one of the major problems affecting the application of the persona and pattern centered processes and the results they produce. In the list of suggested patterns, the tool provides a score, a pattern name and a short description. This information is sufficient to make basic decisions and recall some of the forgotten patterns. However, when a designer needs a closer look at a pattern a simple double click is sufficient to access the detailed information screen. All these simple solutions are aimed at reducing designer's memory load and the effect of experience, specifically related to the use of a particular pattern library.

8.6 Summary

In this chapter, we have explored the supporting environment for P2P process. We have explained the rationale behind P2PMapper tool functionality. For instance, the tool provides the designer with two possible automatic clustering methods. K-means clustering method has been offered as a simple and rapid clustering solution that can produce "rough" grouping results. On the other hand, Sequential Clustering has been designed following process specifications and mimics human persona creation tactics. Since an automated algorithm cannot account for all possible variations in population, we have presented manual grouping alternatives where users can be grouped by simply dragging them close to one another.

Additionally, we have explored possible implementations of Patter Selection process. We have seen that Inference Engine seems to be the most suitable solution that does not require any additional resources and utilizes expert knowledge encoded in form of rules.

Overall, the tool has been designed as a supporting environment that can aid in manipulation of large sets of data while following P2P process. Clustering and Pattern Selection are supported. At the same time, the designer has the choice to intervene in the decision points that rely on analysis based on its expertise. Finally, Pattern Composition is not supported because it is a creative process that can be done by using a pen and a paper.

9. Validation of the process and the supporting tool

Following the implementation of the tool supporting the process presented above, we have conducted a validation study with a goal to:

1. assess the applicability of the P2P process,
2. assess the applicability of the tool within the process and context of an HCI design, and
3. evaluate whether the resulting system is more usable.

We have applied P2P Process, supported by P2PMapper tool with a goal to construct a conceptual design that then has been implemented to construct a fully functional prototype [Javahery et al. 2007(c)]. Then we performed an evaluation of the new design. Similar to the evaluation described in Section 4.3, we have conducted a comparative randomized study using task-based evaluation and open-ended interviews. However, contrary to the NCBI study, we have carried out all predesign empirical studies on other, although similar, applications than the one described below.

In this study, we have used 3D visualization tool called Protein Explorer [2005]. Protein explores is a web-based (Java-script and HTML) application targeted towards biomedical research (see Figure 15). It is generally used for prediction and analysis of complex molecular structures such as proteins, DNA etc. This tool uses a description file called *PDB file* containing all structural information about a given molecule. Moreover, it allows fetching these files from web resources such as Protein Data Bank (PDB) by using PDB ID code that uniquely identifies each molecule.

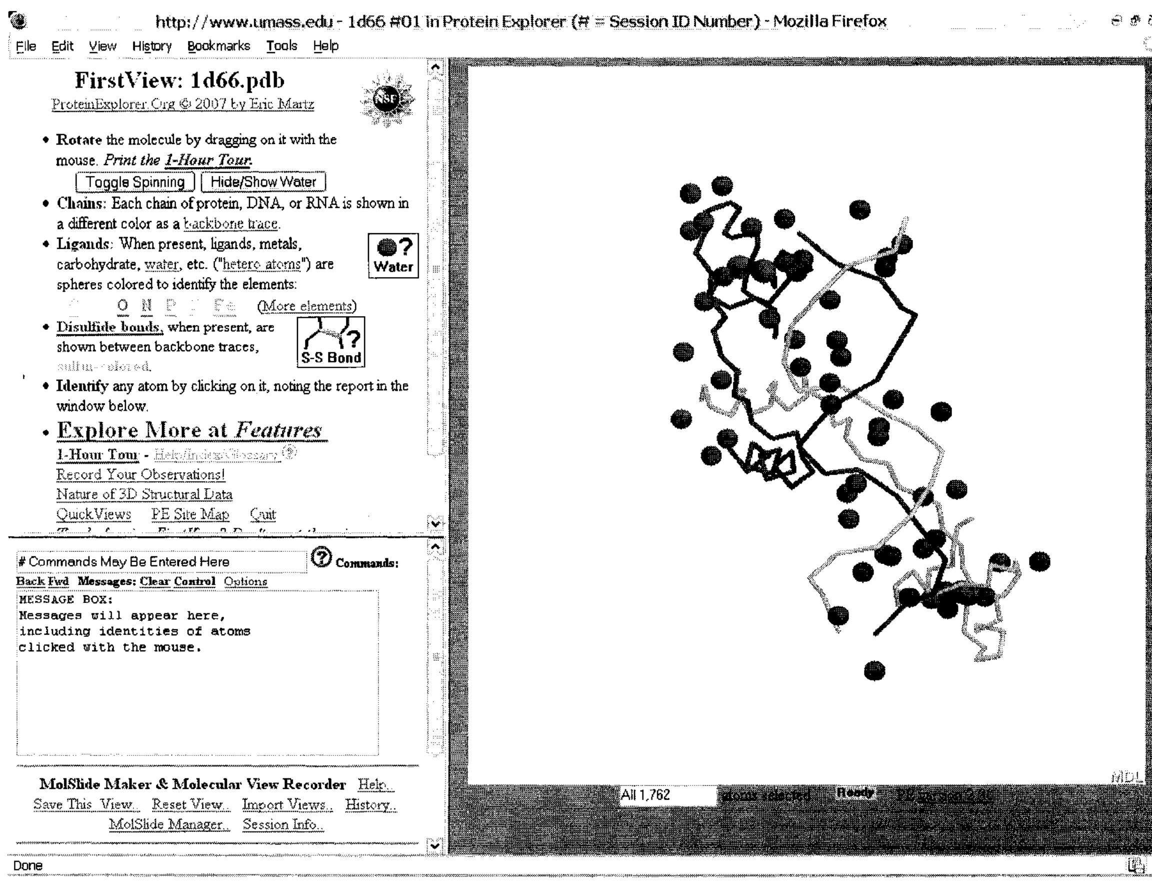


Figure 15: Protein Explorer - First View

During this study, we have applied P2P process as it has been described in previous chapters. Moreover, the process has been supported by P2PMapper Tool functional prototype available at that time.

9.1 Users

Domain and typical users for this study are similar to NCBI. We have selected a sample of 22 users from biology, medical and other related fields. The information collected for each of these users has been used in persona construction. In fact, we carried out usability inquiries in form of field studies and user observations with questionnaire administration on two Bioinformatics visualization tools, Cn3D [2005] and AND-Viewer [Gros et al.

2005]. Initially, each user has been given a short questionnaire in order to provide basic information for his or her profile; however, some information has been collected during user observations. For example, education level, age group and gender were defined based on the questionnaire information; whereas, learning speed variable was defined based on observation results. Finally, we collected information related to user interaction, his/her goals and typical scenarios for a large subset of users.

Table 21: Aggregate description of 22 participants of PE predesign phase.

User Variable	Mean	SD
Age [i]	3.33	0.62
Computer Experience [ii]	2.91	1.15
Domain Experience [ii]	2.09	1.38
Education Level [i]	5.55	0.51
Bio-Informatics Experience [ii]	2.00	1.20
Product Experience [ii]	1.68	1.36
Gender	Male count	Female count
	8	14

i. The scales used are 0 to 6. For more details refer to section 5.1 .
ii. The scales used are 0 to 4. For more details refer to section 5.1 .

9.2 Clustering

Following our process, next step after data collection is clustering. In order to perform clustering, the designer must select a small set of most influential user variables. In order to satisfy this requirement, we have performed the analysis of the results from previously carried activities. We have found that most visible variations of user behaviour and expectations seem to be related to user's *domain experience* and *background.* Therefore, *domain experience* and *background* have been selected as candidates for clustering users.

As we have mentioned earlier, all activities have been performed with the support of P2PMapper tool, when such support was available. However, we must note that initially defined user variables (as presented in the section 5.1) do not include *background* variable. This variable was manually added to user definition as a binary value variable with two possibilities: user has a computer science background or biology background. Moreover, we have also altered the interpretation of *domain experience* variable. In fact, we have assumed that the users belong to three major categories: those who answered Low (0 and 1), Medium (2) and High (3 and 4). In fact, similar to [Enticott et al. 2002] we have considered the responses 0 and 1 as negative rating of the experience, while 3 and 4 were positive.

From "iterative" clustering, we have obtained a set of six clusters (see Figure 16). Then we have analyzed the behaviour of each cluster in comparison with other clusters. Reduced size of information has allowed us to find additional information. More precisely, we have found that the interaction behaviour of biologists with low domain experience was the same as computer scientists. Therefore, one of the groups is sufficient in order to construct a persona. Moreover, when we have re-examined the group of computer scientists with low domain knowledge we found that these users need and tend to use a simpler tool. More precisely, the information contained in scenarios and goals showed that Protein Explorer will not be used by the users of this type. As a result, we have eliminated two of the six groups from the study.

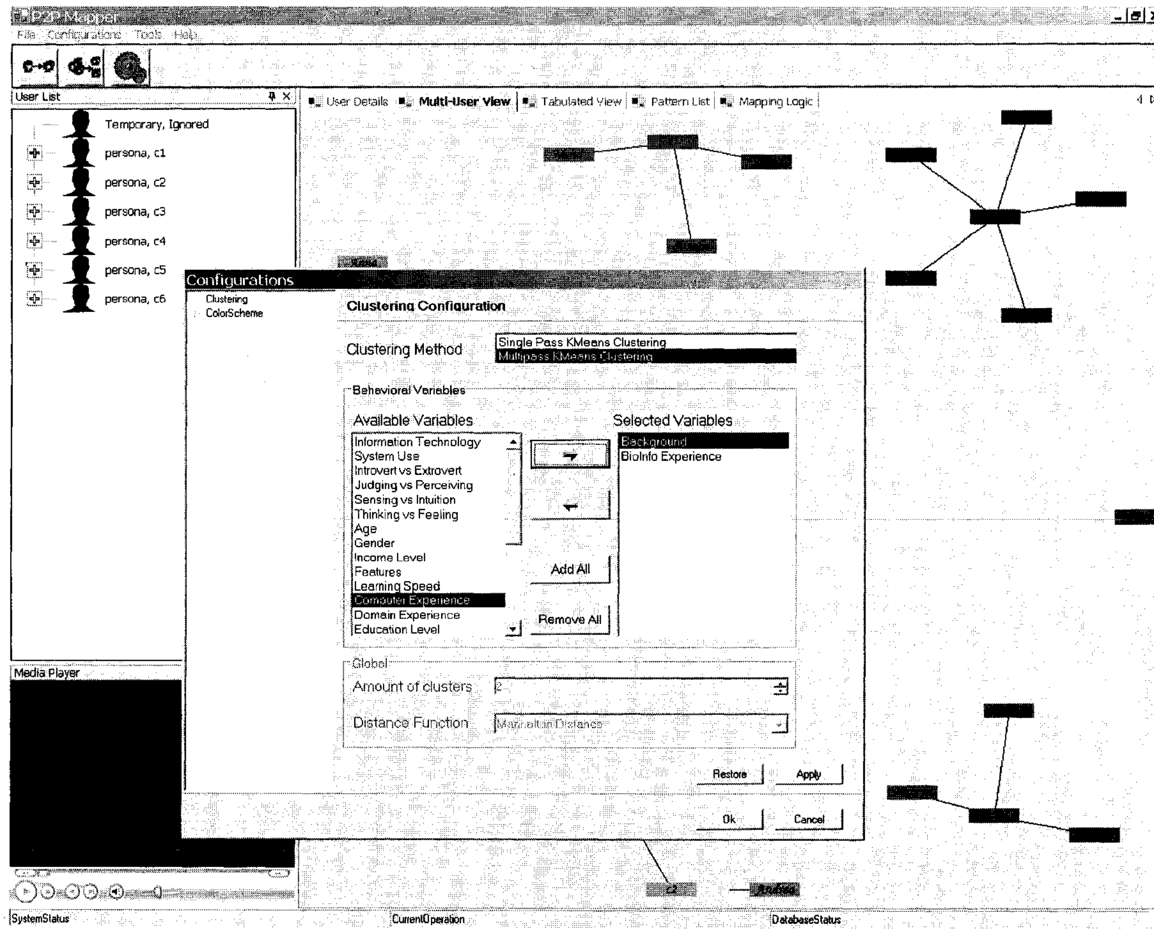


Figure 16: First Iteration Clustering in P2PMapper tool

After analysis of the resulting four groups, we have found that age was an important factor influencing user behaviour. We have found that:

- Older users (45 +) were more anxious and less comfortable when interacting with the system. They also required a higher need for validation of decisions and were feature shy.
- Finally, age variation seemed to be related with users' expectations on the tool support. It was hypothesized that increasing tool support is needed to accommodate reduced learning speed of older users.

Second iteration of clustering was performed on the four remaining clusters with an additional age variable as condition. Although, maximum number of possible clusters was actually 28, some age groups were not present in the user sample. Therefore, the tool produced only eight clusters.

Similar to the method described above, we have compared the behaviour of the users in each cluster with all other clusters. We have found three clusters that represented the behaviour of the remaining five clusters. The resulting clusters C2, C5 and C7 have been again compared and analyzed. However, the analysis did not result in any conclusive finding. This has demonstrated us that the personas should be constructed based on these three clusters. The tool has already provided us with the skeleton structure for each persona: the tool automatically populates the discrete variables of the persona based on the user values in its cluster. In order to bring to life our personas we have constructed from the original user descriptions:

- Martha Aviles, a young bioinformatics professional working in industry (see Figure 17),
- Zhang Hui, a senior Parasitology professor, and
- Sue Blachford, a mature adult and medical practitioner with limited experience in Bioinformatics.

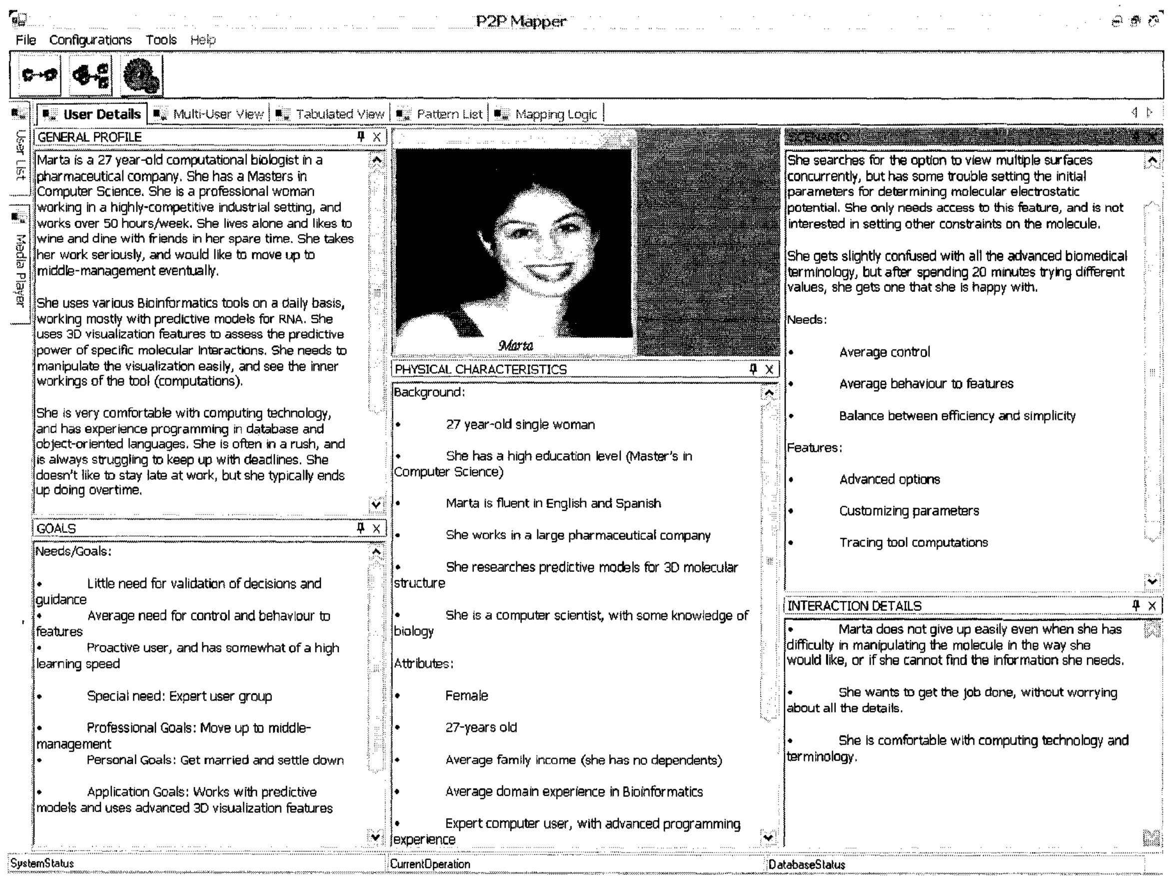


Figure 17: Persona Martha Aviles in P2PMapper tool

9.3 Pattern Selection and Resulting Design

Following the process we have described in previous chapters, we have identified a set of variables that should be used for pattern selection. More precisely, five variables have been identified: (1) *special need*, (2) *age*, (3) *behaviour to features*, (4) *control* and (5) *domain experience.*

Table 22: Selected user variables and values

	Special need	Age	Behaviour to features	Control	Domain experience
Marta Aviles	expert	3	2	2	2
Zhand Hui	colorblind	5	4	4	4
Sue Blachford	novice	4	0	0	1

We have used the tool to compute the scores for all patterns currently available in the library. As we have explained earlier, the tool computes a score to a pattern based on a set of rules and a particular value of an input variable(s). Higher score indicates a higher estimated applicability of the given pattern for the current persona. This process is automatically repeated for each persona.

The results obtained from the tool have been compared in order to assess the possible evolutions of the design. For example, *singe user interface* option and *multiple user interfaces* option were compared. We have noticed that Zhang and Sue had a greatest variation in ranking, when Marta seemed to have a little of both with some differences. Given the environment and the application at hand, we have decided to search for a compromise solution with a single user interface that would accommodate all three personas. Moreover, we had a set of technological limitations related to the accessibility and interaction. More precisely, the new package had to use Java/HTML combination in order to be redistributable and available through web interface. At the same time, the component that reads and displays the molecule has been produced by third parties and is inaccessible to modifications. Based on the above, we have selected 12 patterns (see Table 23).

Table 23: Set of selected patterns for PE design

#	Name	Short description
1	Button Groups	Present related actions as a small cluster of buttons, aligned either horizontally or vertically. Create several of them if there are more than three or four actions.
2	Card Stack	Put sections of content onto separate panels or "cards", and stack them up so only one is visible at a time; use tabs or other devices to give users access to them.
3	Good Defaults	Wherever appropriate, pre-fill form fields with your best guesses of the values the user wants.
4	Legend	Data is encoded in a form of visual objects that are then organized in a scene.
5	Multi-level Help	Use a mixture of lightweight and heavyweight help techniques to support users with various needs.
6	Details on Demand	Hide details of data items and present them on demand as Datatips or in a separate display.
7	Tool Tips	On mouse over an object, give an accurate and short phrase or sentence in close spatial and temporary proximity to the target.
8	Convenient Toolbar	Assist the user to reach convenient and key pages at any time throughout the Website.
9	Action Panel	Instead of using menus, present a large group of related actions on a UI panel that's richly organized and always visible.
10	Command History	As the user performs actions, keep a visible record of what was done, to what, and when.
11	Filter	Provide filtering facilities in order to reduce the number of visual objects displayed or assist the user in finding and focusing on specific items.
12	Reduction Filter	Provide facilities filtering out unwanted items from the display, where the display consists of a number of visual objects.

The selected patterns have been combined in order to create a conceptual design and then an implementation (see Figure 18). It is important to note that we have followed the process described in previous chapters; thus, we have not attempted to fix usability problems directly. In fact, we have attempted to accommodate our personas with the best design possible by reusing best practices encapsulated in patterns. However, we have attempted to keep some formats the same in order to facilitate the transition of users like Zhang to the new design.

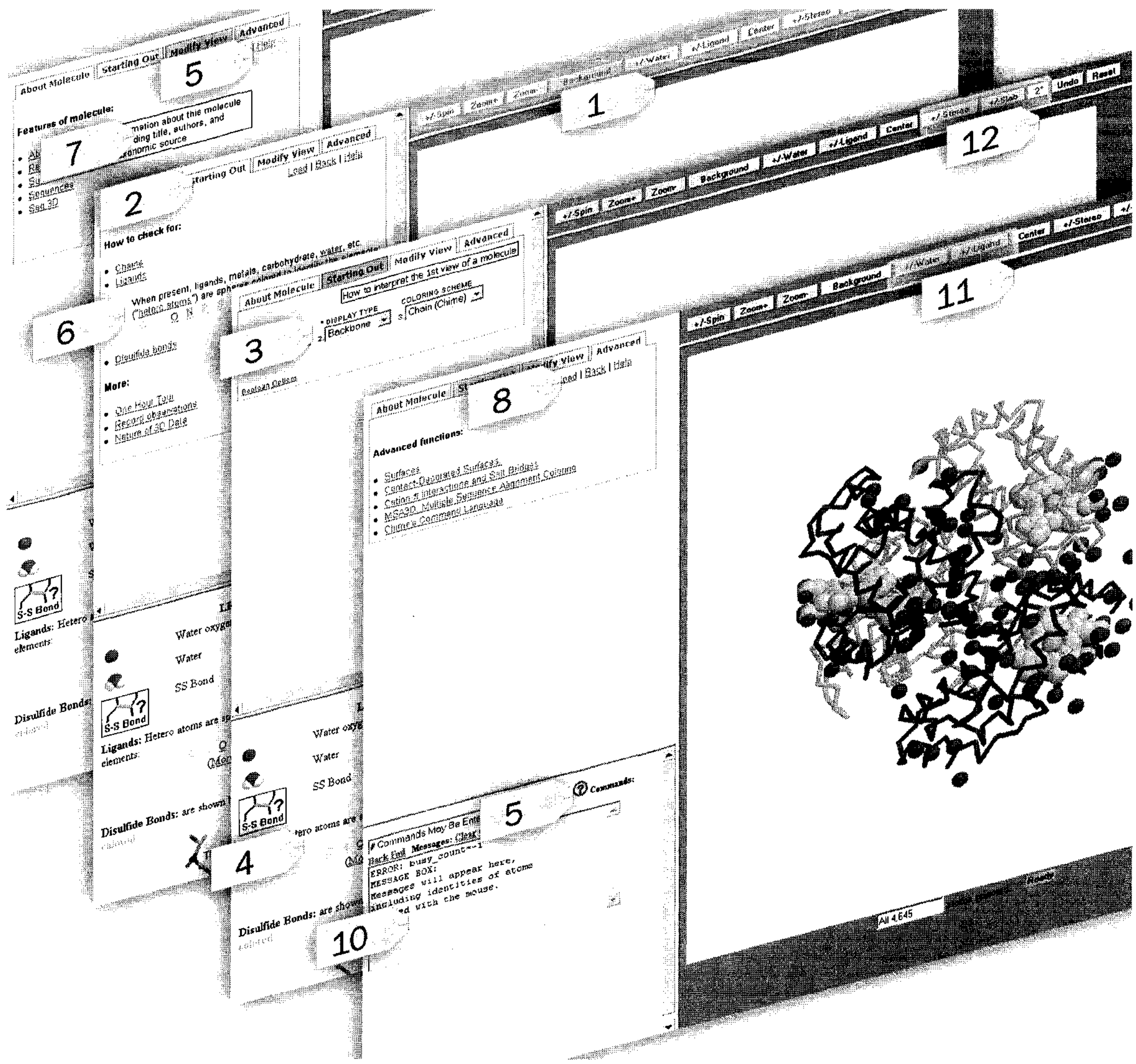


Figure 18: PE Prototype

9.4 Evaluation of PE design

We have performed task-based evaluations and open-ended interviews in order to evaluate the differences in user satisfaction and behaviour with two designs. More precisely, our goal was to test whether new design results in 1) reduced task times, 2) lower failure rates and 3) increase in user satisfaction.

For the testing phase, we have selected a subset of users that have participated in the predesign phase. More precisely, 15 users from biomedical-related field have been selected. Although, some users had experience with 3D visualization tools, none of the selected users had experience with Protein Explorer. Moreover, users were unaware of the application name they will be testing until few seconds before the beginning of the test.

Table 24: Aggregate description of 15 participants of PE testing phase.

User Variable	Mean	SD
Age [i]	3.27	0.55
Computer Experience [ii]	3.00	1.07
Domain Experience [ii]	2.13	1.30
Education Level [i]	5.60	0.51
Bio-Informatics Experience [ii]	1.73	1.03
Product Experience [ii]	1.73	1.22
Gender	Male count	Female count
	4	11

i. The scales used are 0 to 6. For more details refer to section 5.1.
ii. The scales used are 0 to 4. For more details refer to section 5.1.

Since, we had a small user group we have selected *within users* testing protocol. All uses have been given two tasks to perform one on a New Design (Design N) and one on the Original Design (Design O). The tasks have been designed based on information obtained in the predesign phase and with help of biomedical specialist (see Appendix F:). The order in which the users tested the designs varied from user to user in order to reduce the transfer of learning effect. Moreover, at all times during the test users were unaware of the design type (new or old) they are currently using.

Similar to NCBI study, the entire user experience has been recorded. Users facial experience, task times and failure rates have been analyzed in order to extract the information required to satisfy our testing goals. Thus the study has two independent variables (variation of the design type and variation of the design order) and two dependant variables (task time and failure rate).

As we have mentioned earlier, we hoped that our method results in more usable systems. More precisely, following hypothesis have been formulated:

- Proposed process will result in statistically significant improvement of task times when comparing Design O with Design N.
- Process will result in a statistically significant improvement of failure rates when comparing Design O with Design N.
- Effect of transfer of knowledge will be statistically insignificant.

Given the hypothesis above, collected quantitative information was analyzed using ANOVA statistical analysis method. We have performed two-factor with replication test in order to verify the effect that transfer of knowledge had on our results. Moreover, this test was also used to verify if there has been any interaction between two varying factors. The two factors selected for all tests were: 1) the order in which the user tested the designs (O → N or N → O) and 2) the design type tested (O or N).

On the other hand, qualitative information was observed in order to find most common trends and occurrences. For example, the most common comments about the usability of the prototype were: 1) easier to locate information, 2) organization of features and tools seems to follow more closely the process in bioinformatics and 3) the interface is simpler. Moreover, by comparing facial expressions we have found that users testing Design N were calmer. In addition, 13 out of 15 users indicated that they would use the Design N when given the choice of the two. However, one user also indicated that the Design O is better, because it has all the information handy while Design N is too simple. Finally, some users indicated that fonts in Design N were too small.

Although, qualitative analysis has given us an invaluable input, quantitative analysis should clearly reject or confirm our hypothesis.

9.4.1 Task Duration

The results demonstrate that variation of the order in which the user has tested the design (Factor 1) 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, ANOVA test demonstrates that 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: $F=35.71$, $p=3.62E-06$, $\eta^2=0.55$. This demonstrates that there was a statistically significant improvement in task time in Design N when compared to Design O. In fact, variation of the design is the only statistically significant. On average we note an improvement of 52 %.

Table 25: ANOVA two-factor with replication test results for task times

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

Given the results presented above, we can conclude that transfer of knowledge has been effectively reduced to a statistically insignificant level. Moreover, task times have been improved by Design N.

9.4.2 Failure Rates

The test results demonstrated that there was a statistically significant improvement in failure rates in Design N when compared to Design O. In fact, Factor 2 has F=28.03, $p<0.05$ and η^2=0.49.

Moreover, the test has shown that there is no statistically significant interaction between the 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 ($p>0.05$).

Table 26: ANOVA two-factor with replication test results for failure rates

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

Therefore, we can conclude that transfer of knowledge effect has been effectively reduced to statistically insignificant level. Moreover, there was statistically significant improvement in failure rates in case of Design N.

9.5 Tool assessment and improvement

As we have mentioned earlier, P2PMapper tool has been used during the entire design process (where supported). Our goal was to see if the tool is applicable within P2P process. It is important to note that we were not interested in Heuristic evaluation of the tool but rather in advantages and disadvantages of tool usage when following P2P process. Therefore, we have performed an informal interview with the two designers in order to assess possible ameliorations to the tool and gather improvement suggestions.

In general, the designers were satisfied with the tool support. However, some important issues have been mentioned. The designers mentioned that Data Entry is a tedious task that requires a large chunk of time and that there is no clear association between textual description and discrete variables. One of the suggested improvements was to add a facility that allowing a specialist to create a link between a sentence in a textual description and a corresponding discrete variable.

Moreover, they have mentioned that they avoided K-Means clustering because they did not trust it and it seemed to produce unstable results. However, Sequential clustering method was found to be intuitive and suitable for the task. Therefore, experts suggested further exploring fuzzy clustering methods and other statistical analysis tools applicable for both small and large data sets.

Always related with clustering, experts suggested that the proposed interaction model is valid only for small groups of users: smaller than 20. For medium size groups between 20 and 60 users, the relative distance and color guide are not sufficient for an easy association between users. For large group of over 60 users, the proposed view is unsuitable to perform the task. Therefore, further research was suggested to develop a 3D or a different 2D visualization system that will allow for appropriate viewing. Similar to some commercial mathematic and graphical products (MathLab, Maple etc.), it was suggested that each user may be represented as a point and a group of users will look like a cloud in a 3D space (see Figure 19).

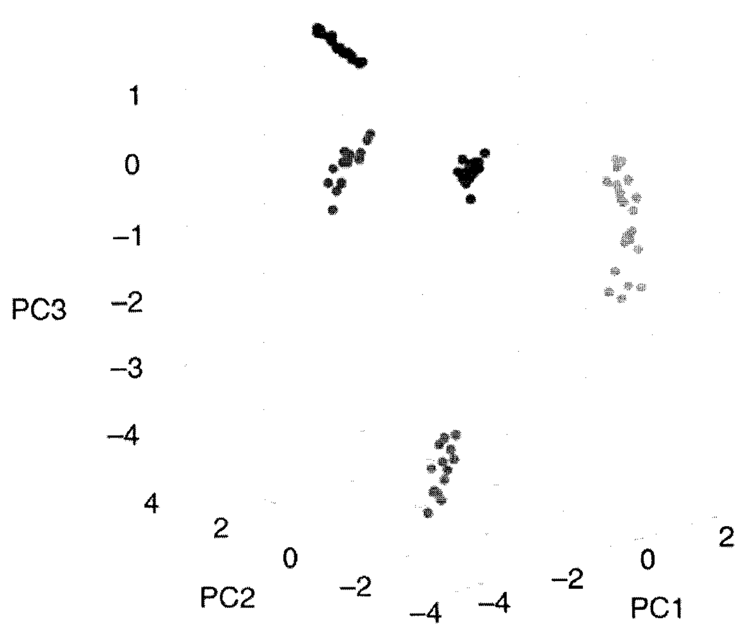


Figure 19: Typical 3D Scatter Plot [Mayer, 2003]

In mapping, the experts found that the trace produced by a tool was not formatted for validation. They have suggested that a trace in form of a sequence of steps executed by Inference Engine is not well suited for score propagation demonstration. Therefore, it was suggested to produce a tree-based representation of the rule execution that will effectively demonstrate the score propagation.

Overall, we have found that the comments obtained from specialist were positive and suggested that the tool has performed according to the expectations: our goal was to construct a functional prototype supporting the process. The proposed improvements were mostly related to the interaction model and GUI components. Since none of the comments was related to support of the process and/or discrepancies between the tool and the process, we can conclude that the tool can be used as a supporting environment for P2P Process.

9.6 Summary

A comparative study was carried out on PE tool. The goal was to verify that application of the process results in numerous improvements. We have found that task times, failure rates and overall user satisfaction have been improved in a new design.

However, some users have complained about colors and font sizes used in the new design. We believe that this indicates at a gap between conceptual design and implementation. In fact, patterns as they are presented now leave a large space for interpretation and implementation. Moreover, there is no mechanism or process guiding the designer when implementing a particular pattern.

We have also found that in some cases we need to modify the original scales for variables, an option that has not been detailed in the process and supported by the tool.

That indicated that further investigation into response to particular scales and their interpretation should be investigated. In fact, such research has been and continues to be performed in psychology and other related fields dealing with questionnaires and scales. However, the designer performing such a change must possess the knowledge required. Therefore, P2P Process must be augmented with clear heuristics or even a sub-process that will effectively guide the designer.

At the same time, we have confirmed that the P2PMapper tool can be used when applying P2P process. However, it was suggested by experts that currently available prototype should be improved by ameliorating interaction and GUI.

10. Conclusion

In this thesis, we have identified some of the commonly occurring problems with the proposed UCD design methods. More precisely, we have considered difficulties with use of Persona and Pattern as part of the design process. Lack of clear format and usage guidelines have been cited as one of the major obstacles.

As a generic solution, we have proposed a set of requirements that need to be satisfied by a UCD process using both patterns and persona. At the same time, we have proposed some modifications to formats of these two artefacts that should improve their applicability, such as a computer readable section in both patterns and persona.

Following the described requirements, we have presented our findings extracted from a Case Study based on a Bioinformatics Website, literature review and other knowledge elicitation techniques. We have detailed both, persona and pattern formats, including their related discrete variables. Additionally, we have presented expert knowledge encoded in the form of rules and used to link patterns to persona.

In the scope of the described findings, we have defined a Persona to Pattern (P2P) process and have described a supporting environment called P2PMapper tool. The proposed process defines a set of steps guiding the specialist from initial set of user descriptions to a conceptual design. It employs both persona and pattern formats as key artefacts in the design. Moreover, the process uses previously gathered expert knowledge to select a set of patterns based on persona description.

Following the process, we have described P2PMapper tool automating some of the most tedious and time-consuming process steps. In fact, the tool proposes semi-automated clustering and manual grouping algorithms. It also offers a fully automated Pattern Scoring module that eliminates the need for score calculation. Additionally, the tool provides easy access to both user (persona) and pattern detailed information reducing the need to remember all the details.

Finally, the proposed process and a tool are evaluated in the scope of the Case Study based on a design of Protein Explorer, a Bioinformatics visualization tool. By comparing to the original design, we find that our prototype resulted in significant improvements in terms of failure rate and task execution times. At the same time, we explore experts satisfaction related with use of the proposed tool as a supporting environment during the P2P process.

While overall results from the research are conclusive and encouraging, a set of research questions have been raised.

First, *persona* format is a good starting point in the way for uniformity and standardization of user descriptions in HCI. However, it would be important to continue the research in this venue. It is necessary to explore the applicability of the proposed format in a set of industrial projects with a goal to assess any possible limitations of the model and their eventual solutions. Particularly, possible research venues could be further detailing variable set and defining a representation for the link between discrete user variable and the associated textual description sentence.

Secondly, our proposed *pattern* format is a slight modification of the current commonly used format. Further investigation is required in order to integrate into pattern artefacts that will help a designer to link it with task models. In the proposed process, we have suggested to filter patterns based on context and task information. Unfortunately, patterns do not provide an explicit description of task type while some of them are applicable to only a restricted task type, such as search or wizard patterns. Specification of task, as in [Javahery 2006] or similar, can provide means for additional manual or automated pattern filtering based on persona task description.

Thirdly, the proposed scoring method is based on rules extracted from expert knowledge accumulated during the elicitation process. However, currently limited knowledge has forced us to make some approximations to define the rules that can be used in scoring. Therefore, a further improvement of scoring through testing with real users is necessary to improve the reliability of the proposed method.

Finally, the proposed tool prototype is a valuable support for a designer when applying P2P process. However, one of the venues that should be explored in relation with the proposition formulated in previous chapter is a combination of scoring techniques. For example, neural network with learning capabilities can be used to enhance the results produced by the tool. More precisely, the resulting score can be combined of the score of the currently proposed set of rules that takes less and less importance as the neural network reliability evolves. Thus, the scoring engine can become a commonly shared component similar to pattern library.

References

Aaker, K. (2004) Marketing Research, 8th edition, Wiley, ISBN 0-471-23057-X.

Accessibility Domain Architecture Team, Accessibility Domain Architecture: Disability / Persona Matrix. Treasury Board of Canada Secretariat, 2003; Available from: <http://www.tbs-sct.gc.ca/fap-paf/documents/accessibility/access18_e.asp #_Toc56226767> [accessed April, 2007]

Accessibility Domain Architecture Team, Accessibility Domain Architecture: Unified Modeling Language Actors: "Personas with Disabilities", Treasury Board of Canada Secretariat, 2003; Available from: http://www.tbs-sct.gc.ca/fap-paf/documents/accessibility/access11_e.asp#_Toc56226747 [accessed April, 2007]

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., and Angel, S. (1977). A pattern language - Towns, buildings, construction. New York, Oxford University Press.

Attwood T.K., Parry-Smith D.J. (1999). Introduction to Bioinformatics. Essex, Addison Wesley Longman Higher Education.

Beyer, H., and Holtzblatt, K. (1998). Contextual Design, Morgan Kaufmann Publishers

Blomquist, A., and Arvola, M. Personas in Action: Ethnography in an Interaction Design Team, NordiCHI, 2002.

Borchers, J.O. (1999) Designing Interactive Music Systems: A Pattern Approach. In Proceedings of 8th HCI International on HCI: Ergonomics and User Interfaces, Volume 1, p. 276 - 280

Carroll, J.M., (2000), Making use: Scenario-based design of human-computer interactions. Cambridge, MA: The MIT Press.

Chapman, C.N., and Milham, R. The personas' new clothes. Human Factors and Ergonomics Society (HFES) 2006, San Francisco, CA. October 2006.

Cn3D (2005), Cn3D 4.1 Homepage [Internet] Available from <http://130.14.29.110/Structure/CN3D/cn3d.shtml> [Accessed October 2006].

Cooper, A. (1999). The inmates are running the asylum. Indianapolis, SAMS Publishing.

Cooper, A., and Reimann, R. M. (2003). About Face 2.0: The Essentials of Interaction Design. Indianapolis, IN: John Wiley & Sons.

Coram T., and Lee J. (1998). Experiences: A Pattern Language for User Interface Design [Internet]. Available from <http://www.maplefish.com/todd/papers/experiences> [Accessed January, 2002].

Courage, C., Baxter, K. (2005) Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques, Elsevier, San Francisco, CA.

Danzico, L. User Profiles, Razorfish, New York (2000) Available from: <http://bobulate.com/documents/scenarios.pdf> [Accessed April, 2007]

Disability Characteristics, 2005 Community Survey, US Census Bureau, 2005. Available from: <http://www.census.gov/hhes/www/disability/2005acs.html> [accessed April, 2007]

Dix, A., Finlay, J.E., Abowd, G.D., and Beale, R. (2003), Human-Computer Interaction, Prentice Hall; 3 edition.

Duyne, D.K. van, Landay, J.A., and Hong, J.I. (2003). TheDesign of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley.

Ehn, P. (1988). Work-Oriented Design of Computer Artifacts. Stockholm: Arbetslivscentrum

Enticott, G., Walker, R.M., Boyne, G.A., Martin, S., and Ashworth, R. (2002) Best Value in English Local Government: Summary Results from the Census of Local Authorities in 2001. Report prepared for the Office of the Deputy Prime Minister, UK, Appendix A. Available from <http://www.clrgr.cf.ac.uk/publications/odpm/report.pdf>

Erickson, T.(2000). Lingua Francas for Design: Sacred Places and Pattern Languages. In Proceedings of Designing Interactive Systems, August 17-19, New York, ACM Press.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA, Addison-Wesley.

Gros, P. E., Férey, N., Hérisson, J., Gherbi, R., and Seffah, A. (2005) Multiple User Interface for Exploring Genomics Databases, Proceedings of HCI International, July 2005, Las Vegas , USA.

Hewett, T.T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, J., Strong, J., and Verplank, W. (1996). ACM SIGCHI Curricula for Human-Computer Interaction, New York, ACM Press. Accessible from: http://sigchi.org/cdg/index.html [Accessed May, 2007]

Javahery, H. (2003) Pattern Oriented Design for Interactive Systems. A Master Thesis, Department of Computer Science, Concordia University, Montreal.

Javahery, H., and Seffah, A.(2002) A Model for Usability Pattern-Oriented Design. Proc. TAMODIA 2002, pp.104-110.

Javahery, H., Seffah, A., and Radhakrishnan, T. Beyond power: making bioinformatics tools user centered. Communications of the ACM 47 (11), 58-63.

Javahery, H., Sinnig, D., Seffah, A., Forbrig, P. and T. Radhakrishnan (2006) Pattern-Based UI Design: Adding Rigor with User and Context Variables, in Proceedings of TaMoDia 2006, Hasselt Belgium.

A) Javahery, H. (2007) Pattern-Oriented UI Design Based on User Experiences: A Method Supported by Empirical Evidence. A PhD Thesis, Department of Software Engineering and Computer Science, Concordia University, Montreal.

B) Javahery, H., Deichman, A., Taleb, M., and Seffah, A.(2007) A User-Centered Framework for Deriving a Conceptual Design From User Experiences, Leveraging Personas and Patterns to Create Usable Designs. In Seffah, A., Gulliksen, J., and Desmarais, M., eds. *Human-Centered Software Engineering: Bridging HCI, Usability and Software Engineering*, Springer (accepted).

C) Javahery, H., Deichmanm A., and Seffah, A.(2007) Incorporating Human Experiences into the Design Process of a Visualization Tool: A Case Study from Bioinformatics, in IEEE-SMC 2007 (accepted).

Jung, C. G. (1971). *Psychological Types*, Collected Works of C.G. Jung Volume 6. Princeton University Press, 1971.

Katz-Haas, R. Usability Techniques: User-Centered Design and Web Development. Usability Interface, Vol 5, No. 1, July 1998. Availble from: <http://www.stcsig.org/usability/topics/articles/ucd%20_web_devel.html> [Accessed May, 2007]

Kim, C., and Kim, J. (2003) A Recommendation Algorithm Using Multi-Level Association Rules. Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03)

Kirakowski, J., and Corbett, M. (1993). SUMI: The Software Measurement Inventory. British Journal of Educational Technology, 24 (5), pp. 210-212.

Lafrenière, D. and Granlund, A. A Pattern-Supported Approach to User Interface Design. Available from UPA 99 Workshop Report, <http://www.gespro.com/lafrenid/Workshop_Report.pdf> [Accessed December, 2002].

Landay, J.A. and Myers, B.A. Sketching Interfaces: Toward More Human Interface Design. IEEE Computer, vol. 34, no. 3, March 2001, pp. 56-64.

Li, N. (2001) Usability Patterns-Assisted Design for Web User Interfaces. A Master Thesis, Department of Computer Science, Concordia University, Montreal.

Maier T., Klepel S., Renner U., and Kostrzewa M. Fast and reliable MALDI-TOF MS–based microorganism identification, first published in Nature Methods, 3 April 2006. Available from <http://www.nature.com/app_notes/nmeth/2006/060328/full/nmeth870.html> [Accessed November, 2006]

MCGrenere, J.(2002) The Design and Evaluation of Multiple Interfaces: A Solution for Complex Software A PhD thesis submitted to Department of Computer Science, The University of Toronto.

Morae (2006) Morae Usability Testing for Software and Web Sites [Internet]. Available from <http://www.techsmith.com/morae.asp> [Accessed Aug, 2006].

Nardi, B.A., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., and Hainsworth, J. Integrating Communication and Information Through ContactMap. Communications of the ACM 45(4), 89-95.

NCBI (2005) National Center for Biotechnology Information [Internet] Available from <http://www.ncbi.nlm.nih.gov/ > [Accessed October 2005].

Nielsen, J. Heuristic Evaluation. In Nielsen, J. and Mack, R.L., eds. Usability inspection methods. New York, NY: John Wiley & Sons (1994).

Ndiwalana, A., Lee, J., Smith, J.L., Wahid, S., Hobby, L., Chewar, C. M., and McCrickard, D. S., "From Personas to Design: Creating a Collaborative Multi-disciplinary Design Environment." In Proceedings of the 11th International Conference on Human-Computer Interaction (HCII '05), Las Vegas NV, July 2005, 10 pages (CD-ROM).

Padda, H. (2003) QUIM map: a repository for usability/quality in use measurement A Master thesis, Concordia University, Department of Software Engineering and Computer Science, Montreal, Canada.

Protein Explorer (2005) FrontDoor to Protein Explorer [Internet], Available from <http://molvis.sdsc.edu/protexpl/frntdoor.htm> [Accessed October 2006].

A) Pruitt, J. and Grudin, J. Personas: practice and theory. Proc. DUX '03, ACM Press, 1-15.

B) Pruitt, J., and Grudin, J., Personas: Practice and Theory. Designing for User Experiences, 2003.

Rönkkö, K. An empirical study demonstrating how different design constraints, project organization, and contexts limited the utility of personas. Hawaii International Conference on System Sciences (HICSS) 2005, Waikoloa, HI. January 2005.

Sommers, A. Using Stories and Fictional Elements to Get the Most out of Your Projects, Business Performance Inc., 2006. Available from <http://learnshareprosper.com/ articles/stories_and_projects.html> [Accessed April, 2007]

Statscanada (2006) Statistics Canada [Internet], Available from < http://www.statcan.ca/> [Accessed September, 2006].

Sutcliffe, A. (2000) On the effective use and reuse of HCI knowledge, *ACM Transactions on Computer-Human Interaction* (7), 2, ACM Press.

Taleb, M., Javahery, H. and Seffah, A. (2006) Pattern-Oriented Design Composition and Mapping for Cross-Platform Web Applications, in Proceedings of DSV-IS 2006, Dublin, Ireland.

Tidwell, J. Common Ground: A Pattern Language for Human-Computer Interface Design. [Internet] Available from <http://www.mit.edu/~jtidwell/ interaction_patterns.html> [Accessed January, 2002].

Tidwell J. UI Patterns and Techniques. [Internet] Available from <http://time-tripper.com/uipatterns/index.php>[Accessed March, 2003].

Tidwell, J. Designing Interfaces [Internet] Available from <http://designinginterfaces.com/> [Accessed October 2006]

Tidwell, J. (2005) Designing Interfaces, O'Reilly, 2005.

Todd, E., Kemp, E., and Philips C. What makes a good User Interface pattern language? AUIC2004, Dunedin, 2004.

UPADE (2004), User Patterns-Automated Design and Engineering [Internet]. Available from <http://hci.cs.concordia.ca/www/hcse/projects/upade/> [Accessed September 2006].

Venturi, G., and Troost J. Survey on the UCD integration in the industry. NordCHI'04, ACM Press, October 2004.

Vetere, F., Balbo, S., Lovett, B., and Khan, A. (2006) PLaSI: Building an Authorative Usability Pattern Collection. Available from <http://smartinternet.com.au/Wizard-AUBcollection> and <http://www.dis.unimelb.edu.au/staff/fvetere/research.html> [Accessed Spring 2006]

Welie, M.V. Interaction Design Patterns. [Internet] Available from <http://www.welie.com> [Accessed March, 2003].

Welie, M.V., Van der Veer, G.C., and Eliëns, A.(2000) Patterns as Tools for User Interface Design. Proc. of the International Workshop on Tools for Working with Guidelines, October 7-8, Biarritz, France.

Welie M.V., and Trætteberg, H. (2000) Interaction Design Patterns in User Interfaces, in Proceedings of PLOP'2000, Monticello, Illinois, USA.

Wilkins, B. (2003) MELD: A Pattern Supported Methodology for Visualization Design, a PhD thesis submitted to The University of Birmingham, School of Computer Science.

Yacoub, S., and Ammar, H. (2003). Composition of Design Patterns, Addison Wesley Professional, ISBN 0-201-77640-5, 416 pages, Hardcover, Germany.

Zimmer, W. (1995). Relationships between design patterns. New York, NY, USA. ACM Press, pp 345-364

Appendix A: Persona Set (NCBI study)

Table 27: Initial set of persona (NCBI study)

P1	**Donna Smith** (Young adult, Student, Basic)	P2	**Xin Li** (Mature adult, Industry, Advanced)	P3	**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 28: Selected descriptions of final persona set

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
She recently started doing bioinformatics-based research, and has only been accessing the NCBI site for 6 months	He has been accessing the NCBI site for 2 years now, and is very familiar with tools related to his research
She is still unfamiliar with all the menu options and functions and often needs guidance	English is his second language, and he is not always comfortable with spelling
She is still learning about the NCBI site, and actively reads "General NCBI Information" and "About NCBI"	He uses the NCBI site for specific tasks, such as secondary structure prediction for proteins and wants to save his results
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	Likes to limit his searches to specific species and doesn't have patience to go through a long list of possibilities
She gets lost looking for information after advancing more than 3 layers, and needs to go back to a safe place	Likes to know about recent discoveries and advances in the field

Appendix B: Commonly Used Pattern Formats

Table 29: INTERACT' 99 Pattern Format. From The Pattern Gallery (Sally Fincher, 2000)

Pattern	
Name	Should encapsulate the pattern's intent. Ideally, short and pithy
Sensitising example	A concrete example of implementation of the pattern. In Alexander, the photograph conveys this example of implementation, in GoF patterns it is the code sample. We took it that the purpose of these components is to sensitise the reader to the application of the pattern. "In looking at the photograph, a reaction is invoked. The intention is that the reaction is favourable-"Wow, that's good. I'd like to live there"-and from that point the reader is sensitised so that the information that the rest of the pattern contains becomes more accessible, more useful in a specific implementation". (Fincher, JCMST, 18(3)) Our expectation was that for UI patterns, this example would most likely be a photograph or a screenshot of an interface, or (depending on medium) possibly a video of a task being accomplished.
Problem Statement	Normally expressed as a conflict between forces
Body	Textual description
Solution Statement	Tells you what to do, not how to do it
Technical representation	We considered this to address the audience of HCI experts, rather than users, or experts in other domains (i.e. the audiences most receptive to the sensitising example). It differs from the sensitising example in that it should represent the solution less impressionistically and with less potential for ambiguity. A possible medium might be UML
Related Patterns	Other patterns which either: are peer to this one, enhance this one or complete this one.
Attribution	

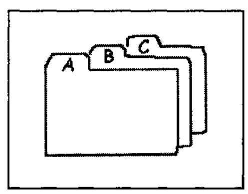
Identification	**Pattern Name**	**STACK PAGE**
	Category	Product-Oriented Patterns→ Structural Patterns→ Page Managers Patterns
Context_Use	User	Novice and Expert
	Task	- Contents are grouped into categories. - No obvious hierarchical structure exists amongst those categories or topics.
	Workplace	Web page
Usability_Problem	- All the information is clearly grouped. - The user can easily navigate the information through switching the tabs of the working spaces. - The user can only access one category each time.	
Usability_Factor	Factor	Effectiveness
	Criteria	Understandability
Example		
Design_Principle	Conceptual Model	Constraints, Affordance, Mapping
	Dialog Style	WYSIWYG
Design_Solution	- Utilize several surfaces stacked together to group information and label them by the name of the categories. - Locate the navigation area at the top if the number of surfaces is less than 8; otherwise, put the navigation area on the left side.	
Implementation		
Usability_Patterns	Super-ordinate	Sequential, Hierarchical, Grid, Composite
	Sub-ordinate	Executive Summary, On Fly Description, Browsing Index
	Neighboring	Focus Page, Tiled Page
	Competitor	Focus Page, Tiled Page
Reading	- "Stack of Working surfaces" in Common Ground. - "Navigating spaces" in Amsterdam Collection - "Garden of Windows" in Experience.	

Figure 20: A Stack Page pattern from UPADE [2002]

3.2 Interaction Design Pattern: Incremental Revealing
Context: Decide how to unfold contents and features of an interactive system so that it conveys a *Simple Impression* to *Attract Users* and *Engages Users*.
Forces: A simple impression is important to make a system look non-intimidating and inviting, especially for novices. **But:** To keep users engaged, the system needs to convey its depth of features and contents as well.
Solution: Initially, only present a concise and simple overview of the system functionality. When the user actively shows interest in a certain part of this overview, offer additional information about it, revealing in successive stages what lies behind the initial presentation.
Examples: Desktop GUIs hide menu entries in menu bars until the user selects a menu. *WorldBeat* has a simple main selection screen with only names and icons for composing, conducting, etc.; when the cursor is over an item, a short text explains it; when it is selected, the system switches to the new page.
Consequences: Incremental revealing is easier when the contents have a *Flat & Narrow Tree Structure*. To show what lies behind a user interface object, use *Dynamic Descriptors* (as in Mac OS Balloon Help, or Windows ToolTips).

Figure 21: Pattern Example as proposed by Borchers [1999]

Overview Plus Detail	
[Image] From Mail for OS/X	
Use when:	Place an overview of the graphic next to a zoomed "detail view." As the user drags a viewport around the overview, show that part of the graphic in the detail view.
Use When:	You show a data set drawn as a large information graphic -- especially an image or a map. You want users to stay oriented with respect to the "big picture," but you also want them to zoom down into the fine details. Users will browser through the data, inspect small areas, or search for points of interest. High-level overviews are necessary for finding those points of interest, but users don't need to see all available detail for all data points at once -- zooming in on a small piece is sufficient for getting fine detail.
Why:	It's an age-old way of dealing with complexity: present a high-level view of what's going on, and let the user zoom from that view into the details as they need to, keeping both levels visible on the same page for quick iteration. Edward Tufte uses the terms "micro and macro readings" to describe a similar concept for printed maps, diagrams, and other static information graphics. The user has the large structure in front of them at all times, while being able to peer into the small details at will: "the pace of visualization is condensed, slowed, and personalized." Similarly, users of Overview Plus Detail can page methodically through the content, jump around, compare, contrast, move quickly, or move slowly. Finally, the overview can serve as a "You are here" sign. A user can tell at a glance where they are in the context of the whole data set by looking for the viewport on the overview.
How:	Show an overview of the data set at all times. It can be an inset panel, as in the example at the

	top of the pattern. It could also be a panel beside the detail view, or even another window, in the case of a multiwindow application like Photoshop (see the examples below). On that overview, place a viewport. They're usually red boxes, by convention, but they don't have to be -- they just need to be visible at a glance, so consider the other colors used in the overview panel. If the graphic is typically dark, make it light; if the graphic is light, make it dark. Make the viewport draggable with the pointer, so users can grab it and slide it around the overview. The detail view shows a magnified "projection" of what's inside the viewport. The two should be synchronized. If the viewport moves, the detail view changes accordingly; if the viewport is made smaller, the magnification should increase. Likewise, if the detail view has scrollbars or some other panning capability, the viewport should move along with it. The response of one to the other should be immediate, within a tenth of a second (the standard response time for direct manipulation).
Examples:	**[Image]** From Photoshop Photoshop places the image canvas (the "detail view") on the left, and the overview on the right. The Navigator window shows a view of the whole image, with a red box showing the size and scroll position of the image's canvas window. **[Image]** From the MATLAB Signal Browser In the signal-processing example shown here, the overview panel is on the bottom of the window. The "You Are Here" aspect is particularly important for users who deal with long, complex signals.

Figure 22: Overview Plus Detail pattern as proposed by Tidwell [2006]

Appendix C: Simplified Access patterns

The examples included in the patterns below are part of the City of Montreal website, at http://ville.montreal.qc.ca. After a summit of Montreal held in June 2002, City of Montreal, started an initiative of universal access for people with disabilities. In short, universal access principle is a possibility for a person with limitations and disabilities, visual, auditives, intellectual or others, to participate in the life of the community or to use available products and service autonomously. 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, the site "Access Simplifie" has been constructed for people having intellectual incapacities or major reading or language problems. This is a unique experience worldwide:

City of Montreal is the first municipality in the world to construct such a web site adapted for the citizens with disabilities. This site has been build with help and support of DEFI Apprentissage de l'Université de Montréal, working in research for 20 years, and Comité régional des associations pour la déficience intellectuelle (CRADI). [Montreal 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 work is based on guidelines referenced in [Montreal 2006].

Table 30: Simplified Text Pattern

Title	Simplified text
Context	The user has difficulties in reading and understanding information read.
Problem	How to reduce the complexity of the communicated information without necessarily impeding on 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 a simple and direct language; – Present a single main idea in a phrase; – Avoid abbreviations; – Have clear and logical structure. Simplified text pattern offers information in a condensed easy to read and understand form. The work is based on the rules and directives published in the guide "Le savoir simplifier" produced by ILSMH European Association. Authors of this guide propose to always use a simple and direct language, to present a single main idea per phrase, avoid abbreviations and have clear and logical structure. By applying these rules we fight exclusion from information feeds.
Examples	 from [http://ville.montreal.qc.ca]
Related Patterns	Neighbouring: Alternative Orthography, Audio Communication.

Table 31: Alternative Orthography pattern

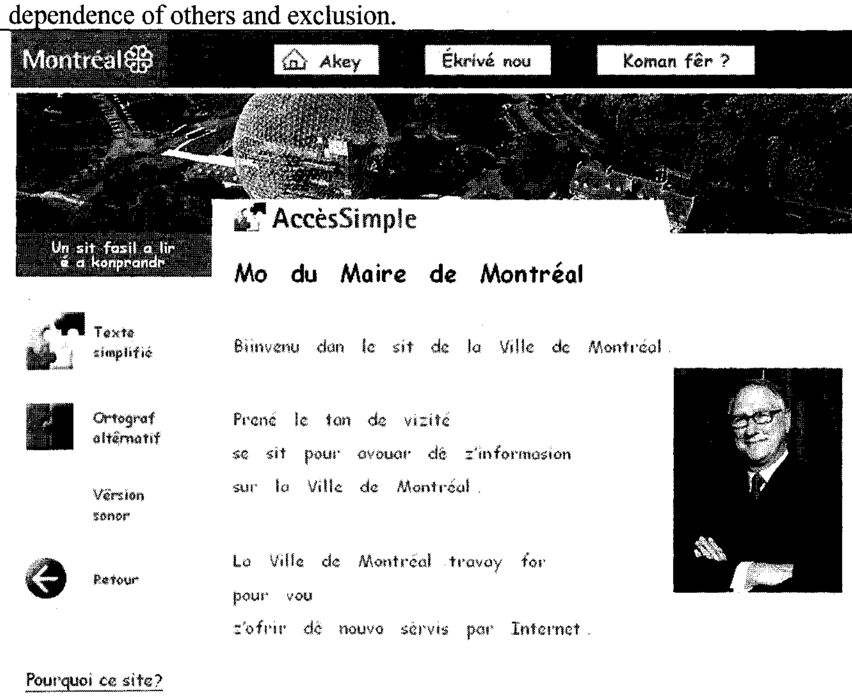
Title	Alternative orthography
Context	The user has difficulties in reading and understanding information read because of the limited intellectual capacity.
Problem	How to reduce the complexity of the medium used (language) to communicate information without necessarily impeding on the content.
Forces	May require a simplified written communication
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" process. Use only one combination of letters to produce the same sound. This pattern is addressed towards a portion of population that has limited intellectual capacities. In addition to simplification of the content, Alternative Writing reduces the complexity of the writing. This writing style is based on a stable relation between letters and sounds. Alternative Writing uses only 35 relations when conventional french language counts more than 4000. Attention, Alternative Writing is not a new way to write in french. It is an alternative way, similar to brail, destined for the people whom have been placed in the situation of illiteracy, dependence of others and exclusion.
Examples	 from [http://ville.montreal.qc.ca]
Related Patterns	Neighbouring: Simplified Text, Audio Communication.

Table 32: Audio Communication

Title	Audio Communication
Context	The user is not capable of reading and understanding information read because of the limited intellectual capacity.
Problem	How to present information in a format understood by the user.
Forces	Presents information in simple well organized format Does not require any knowledge of written communication.
Solution	Allow for audio communication of information present on a screen. The information should be organized and presented in audio format allowing for easy understanding and assimilation of information. When possible, provide means for audio interaction between system and user. Sound offers means of complementary communication of all the texts present in the previously described formats.
Examples	N/A
Related Patterns	Neighbouring: Simplified Text, Alternative Orthography.

In order to demonstrate the resulting design let us compare the home pages of the City of Montréal website. On the image below on left is the home page targeted towards general public and on the right towards users with disabilities.

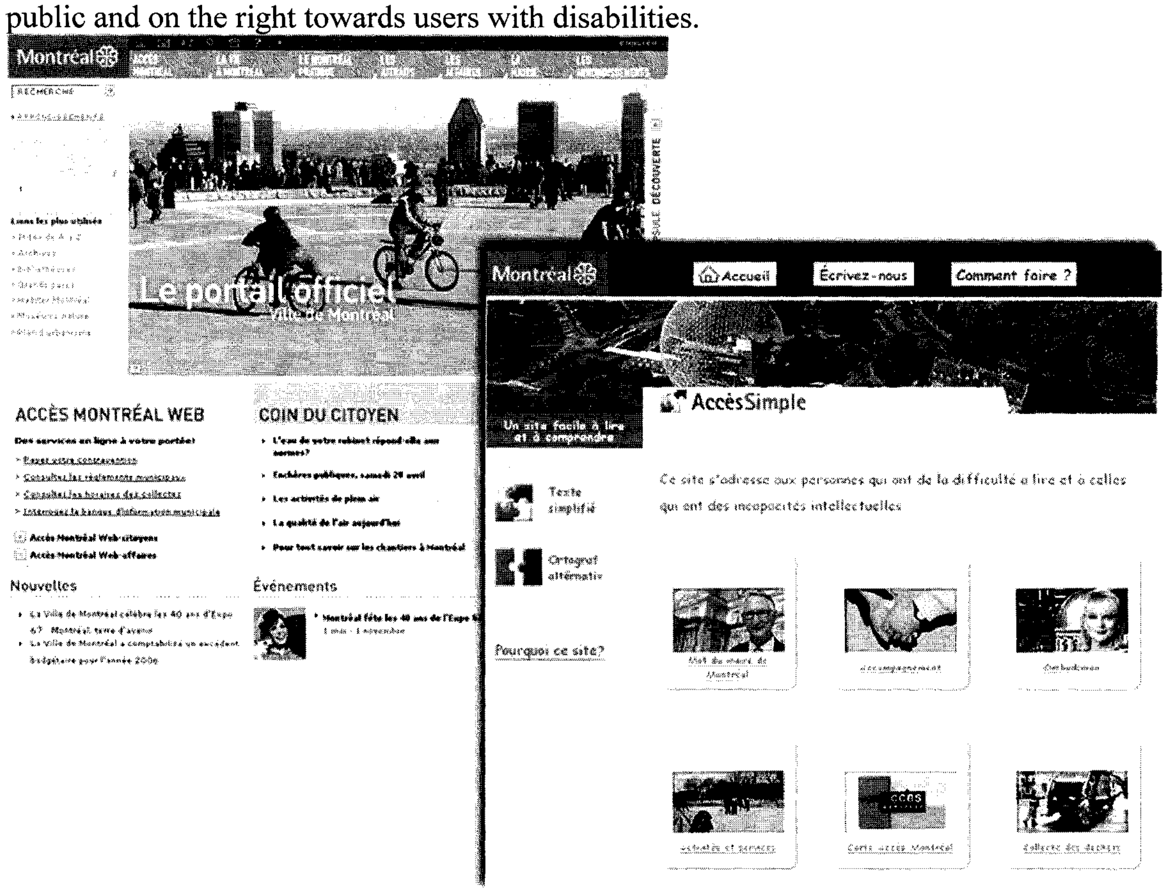


Figure 23: Comparing resulting designs for City of Montreal Home page

Appendix D: Personas formats

Following format has been adopted from a template provided by Adele Sommers from Business Performance Inc. [2006]

Persona Profile (First Page)

Name	**Personal Story**
Age	Picture and Set of quotes describing relevant situations, thoughts etc.
Gender	
Education	
Occupation	
Area of residence	
Income	
Home life	
Ethnic/cultural background	
Hobbies and recreation	
What problems keep him or her up at night?	
Personal attributes	
Goals	

Persona Profile (Second Page)

[Picture goes here]	**Goals (3)**	**Sample Life Scenarios (3)**	**Needs**	**Ideal System Features**
Thumbnail Sketch A short description of the subject in the point form including a quote representing the main behavioural trait.	Wants, plans, whishes etc.			

Following format was developed at Razorfish architecture group in New York by Liz Danzico [2000]. It has a well defined organization and a simple to view one page format (it was split into two pages to accommodate printing requirements).

1 *the learner*

Personal Profile

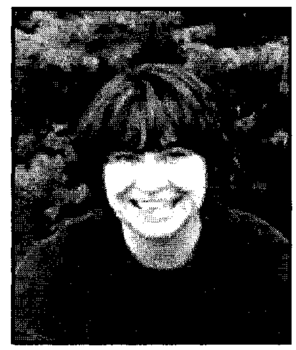
Sandy's work life is hectic, and "learning about investing" has been just another item on her to-do list. She has money in a 401(k) plan, but her conservative investment picks have been under-performing the market rather seriously. She recently received a $10,000 bonus at work, and for the first time, she doesn't need the money to pay off bills.

Sandy is mainly interesting in building wealth. She's concerned about having enough money to retire comfortably, and she doesn't have any rich relatives to count on. Recently married, she and her husband are starting to talk about buying a home and starting a family. As a teacher in a private school, her husband has a moderate income, but a rather meager pension plan.

Before making any major decision, Sandy likes to thoroughly research and understand her choices. She is conservative with money and has a low risk tolerance. While Sandy is concerned about making smart investment choices, she has been feeling lately that she's missing out on the hot stock market.

With $10,000 to invest, Sandy feels that it's time to start investing more seriously. She doesn't want investing to take over her life, but she would like to learn enough to feel comfortable making decisions. She'd be most comfortable having a professional manage her money, but she doesn't think she has enough money to justify hiring someone. And she doesn't expect her small portfolio to get much attention from a professional.

Sandy Kelchwick



"I'm interested, but I'm not sure how to go about it."

background

- 30-year-old, married, woman.
- Received a BA at Cooper Union and a masters in architecture at Columbia University.
- Works in a 20-person architecture studio in Manhattan.
- Intermediate Internet user, has fast connection to PC at work and slow dial-up line on an IMac at home
- Has about $20,000 in a 401(k) plan her employer set up for her
- Wants to learn how to better manage her investments.
- Knows a little about mutual funds, and would like to learn about stock investing

attributes

- Younger
- Female
- Less wealthy
- More experienced with computers and the Internet
- Inexperienced with investing

customer needs

- Simplicity and ease-of-use
- Guidance
- Learning tools
- Planning tools
- Help moving from big picture to specific actions
- Proactive communication
- Service
- Validation of decisions

r a z o r f i s h

Scenario	Needs	Feature	Behavior
Sandy wants to learn enough about investing to understand how to pick the right stocks, and how to decide when to buy and when to sell. She is particularly interested in socially responsible stocks. She picks up a book called "Investing for Dummies," and visits the Motley Fool website after hearing about it on the radio.	• Education • Guidance	• Monthly Newsletter • Market Hilites	*Sandy goes to her MySchwab page and sees that the Monthly Newsletter provides xx.* *Sandy tries looking up a stock in Quotes & Research. She notices that on this page, she can subscribe to Market Hilites. She feels that this would be a good overview of what's happening in the market and a complement to her learning.*
Based on her research, Sandy decides to invest part of her money in index funds, and use the rest to buy individual stocks. She'll identify a few stable, well-established and socially responsible companies.	• Guidance • Validation of decisions	• Monthly Newsletter • Full Closing Bell	*Sandy finishes the Retirement Planner and notices the Monthly newsletter. She applies.*
Because Sandy has very little time to actively trade, her strategy is to monitor these companies' stocks and buy them when there is a dip in prices. She plans to watch them over time, but hold onto them for the long term.	• Guidance • Validation of decisions	• Price • Volume • Early Closing Bell	*Sandy adds these stocks to her Watch lists. While on this page, she realizes that she can sign up for Price and Volume Alerts so she won't have to manually monitor these stocks.* *After placing a trade, Sandy notices that she can receive an Early Closing Bell. She can just take a glance over that once during the day to check out the unofficial prices.*

Following format was used by Microsoft in a few large-scale projects and has been reported in a book by Courage and Baxter [2005]. It is particularly interesting to note that this format defines some new aspects, such as International Considerations and Technology Attributes. While not clearly identified in the format, each aspect should have a link to supporting data allowing for review, evolution and tracing back to the source of the decision.

Overview – Patrick Blakeman (the consultant)
Get to know Patrick, his business and family.
A Day in the Life
Follow Patrick through a typical day.
Work Activities
Look at Patrick's job description and role at work.
Household and Leisure Activities
Get information about what Patrick does when he's not at work
Goals, Fears and Aspirations
Understand the concerns Patrick has about his life, career, and business.
Computer Skills, Knowledge, and Abilities
Learn about Patrick's computer experience.
Market Size Influence
Understand the impact people like Patrick have on our business.
Demographic Attributes
Read key demographic information about Patrick and his family.
Technology Attributes
Review Patrick's perspective on technology, past and future.
Communicating
Learn how Patrick keeps in touch with people
International Considerations
Find out what Patrick is like outside the country (U.S.)
Quotes
Hear what Patrick has to say.
References
See source materials for this document.

Appendix E: P2PMapper Tool Data Format

Following the requirements described in the process, we have developed a user and a pattern description. In the description of the data model we have established that the descriptions of the user and pattern should contain a textual (essentially qualitative) and computer readable (quantitative) information.

Moreover, we have established that the data model must be flexible and should provide multiple means for adaptation based on domain of application and evolution of the field. However, we suggested that most of the changes should occur in the quantitative section of the description.

Both requirements, suggest that the medium used for storing information on users and patterns should be flexible, allow relatively easy modification, be commonly used and accepted and allow for definition of custom data items. Therefore, our choice is XML as a medium for representation of users and patterns. To fulfill our needs we have used an XML Schema Definition provided by Microsoft. However, these tags can easily be defined with any other DTD.

Appendix E.A: User definition as XML Schema

It is essential to have a description of a user where each value and each possibility for an attribute is clearly defined and identified. This ensures that the format can be used understood and even extended following the same principles.

Textual section has also been used in order to define a set of supporting elements such as reference to a picture and a media file for a given user. In general, textual description section can be identified by the presence of a set of elements with type string. In description below, it is the first block of elements.


```
<xs:element name="users">
    <xs:complexType>
        <xs:sequence>
        <xs:element name="id_key" type="xs:string" msdata:AutoIncrement="true" msdata:AutoIncrementSeed="0"
        />
        <xs:element name="lname" type="xs:string" default="unknown" />
        <xs:element name="fname" type="xs:string" default="unknown" />
        <xs:element name="picture_key" type="xs:string" default="unknown.jpg" />
        <xs:element name="General Profile" type="xs:string" default="" minOccurs="0" />
        <xs:element name="persona_key" type="xs:string" default="-1" minOccurs="0" />
        <xs:element name="ispersona_key" type="xs:boolean" default="false" minOccurs="0" />
        <xs:element name="Physical characteristics" type="xs:string" default="" minOccurs="0" />
        <xs:element name="Cognitive style" type="xs:string" default="" minOccurs="0" />
        <xs:element name="Goals" type="xs:string" default="" minOccurs="0" />
        <xs:element name="Scenario" type="xs:string" default="" minOccurs="0" />
        <xs:element name="Function" type="xs:string" default="" minOccurs="0" />
        <xs:element name="Interaction details" type="xs:string" default="" minOccurs="0" />
        <xs:element name="ignore_key" type="xs:boolean" default="false" minOccurs="0" />
        <xs:element name="mediafile_key" type="xs:string" default="" minOccurs="0" />

            <xs:element name="Knowledge_and_Experience">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Education Level" type="mstns:Value9" use="optional" />
                    <xs:attribute name="Domain Experience" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Literacy" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Computer Experience" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Product Experience" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Linguistic Ability" type="mstns:Value6" use="optional" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Demographics">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Age" type="mstns:Value8" use="optional" />
                    <xs:attribute name="Income Level" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Gender" type="xs:boolean" use="optional" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Attitude_and_Motivation">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Information Technology" type="mstns:Value6" use="optional" />
                    <xs:attribute name="System Use" type="mstns:Value6" use="optional" />
```

```
                </xs:complexType>
            </xs:element>
            <xs:element name="Cognitive_Style">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Introvert vs Extrovert" type="mstns:Value3" use="optional" />
                    <xs:attribute name="Judging vs Perceiving" type="mstns:Value3" use="optional" />
                    <xs:attribute name="Thinking vs Feeling" type="mstns:Value3" use="optional" />
                    <xs:attribute name="Sensing vs Intuition" type="mstns:Value3" use="optional" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Learning_Style">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Auditory" type="mstns:Value3" use="optional" />
                    <xs:attribute name="Visual" type="mstns:Value3" use="optional" />
                    <xs:attribute name="Kinesthetic" type="mstns:Value3" use="optional" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Needs">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Guidance" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Validation of Decisions" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Control" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Initiative Taking" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Learning Support" type="mstns:Value6" use="optional" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Interaction_Style">
                <xs:complexType>
                    <xs:sequence />
                    <xs:attribute name="Features" type="mstns:Value6" use="optional" />
                    <xs:attribute name="Learning Speed" type="mstns:Value6" use="optional" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Appendix E.B: Pattern definition as XML Schema

While this was not our primary goal, the resulting format can be converted into most commonly used pattern formats such as PLML.

```
<xs:element name="patterns">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="id_key" msdata:AutoIncrement="true" type="xs:string" msdata:Ordinal="9" />
```

```
            <xs:element name="Relationships" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Relationship Type" type="rel_type" minOccurs="0"
                        msdata:Ordinal="0" />
                        <xs:element name="cid_key" type="xs:string" minOccurs="0" msdata:Ordinal="1" />
                    </xs:sequence>
                    <xs:attribute name="id_key" type="xs:int" use="prohibited" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Information" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="primary criteria" type="xs:string" minOccurs="0"
                        msdata:Ordinal="0" />
                        <xs:element name="secondary criteria" type="xs:string" minOccurs="0"
                        msdata:Ordinal="1" />
                        <xs:element name="type" type="pattern_type" minOccurs="0" msdata:Ordinal="2" />
                        <xs:element name="task type" type="task_type" minOccurs="0" msdata:Ordinal="3"
                        />
                    </xs:sequence>
                    <xs:attribute name="id_key" type="xs:int" use="prohibited" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Special_Needs" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Group Type" type="special_group" minOccurs="0"
                        msdata:Ordinal="0" />
                    </xs:sequence>
                    <xs:attribute name="id_key" type="xs:int" use="prohibited" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="Title" type="xs:string" default="unknown" />
        <xs:attribute name="Context" type="xs:string" default="unknown" />
        <xs:attribute name="Problem" type="xs:string" default="unknown" />
        <xs:attribute name="Forces" type="xs:string" default="unknown" />
        <xs:attribute name="Solutions" type="xs:string" default="unknown" />
        <xs:attribute name="Examples" type="xs:string" default="none" />
        <xs:attribute name="Related patterns" type="xs:string" default="none" />
        <xs:attribute name="Type" type="xs:string" default="unknown" />
        <xs:attribute name="Short" type="xs:string" default="unknown" />
    </xs:complexType>
</xs:element>
```

Appendix F: Patterns and related pattern criteria

Table 33: Examples of Patterns and related pattern criteria

Pattern Name	Short Description	Primary Criteria	Secondary Criteria
Tree Table	Use a Table and indented outline structure to demonstrate hierarchical dependencies.	Logical Org.	
Action Panel	UI panel for related actions instead of menus.	Logical Org.	Accelerators
Details on Demand	Display item details in a separate window.	Min. Design	Feedback
Sequential	Organize Pages in a sequence.	Navigation	Logical Org.
Corner Treatments	Use corner Treatments instead of right angles.	Appeal	Consistency
Multi-level Help	Mixture of help techniques for varying needs.	Feedback	Error Prev.
Row Striping	Alternate the background color of rows using two similar shaded.	Logical Org.	Error Prev.
Few Hues, Many Values	Choose at most three color hues to use in the interface.	Min. Design	Consistency
Button Groups	Present related actions as a small cluster of buttons.	Logical Org.	Min. Design

Appendix G: Tasks for 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).

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).